PCT

WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 7 : G06F 17/60

A1

(11) International Publication Number: WO 00/46705

(43) International Publication Date: 10 August 2000 (10.08.00)

(21) International Application Number: PCT/GB00/00297

(22) International Filing Date: 1 February 2000 (01.02.00)

(30) Priority Data:
9902310.3 2 February 1999 (02.02.99) GB

(71) Applicant *(for all designated States except US)*: LOMBARD RISK MANAGEMENT PLC [GB/GB]; 13th floor, 21 New Fetter Lane, London EC4A 1AJ (GB).

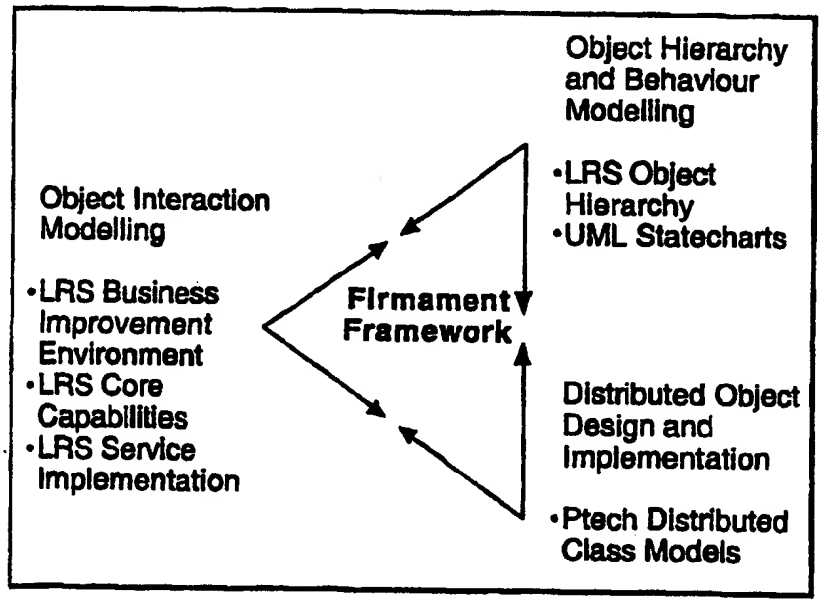
(72) Inventors; and
(75) Inventors/Applicants *(for US only)*: WISBEY, John, Michael [GB/GB]; 9 Eaton Mansions, Cliveden Place, London SW1W 8HE (GB). ERL, Miodrag, Joseph [HR/GB]; 21 Ladywood Mead, Leeds, West Yorkshire LS8 2LZ (GB).

(74) Agent: GILL JENNINGS & EVERY; Broadgate House, 7 Eldon Street, London EC2M 7LH (GB).

(81) Designated States: AE, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CR, CU, CZ, DE, DK, DM, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, US, UZ, VN, YU, ZA, ZW, ARIPO patent (GH, GM, KE, LS, MW, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

Published
With international search report.

(54) Title: BUSINESS OPTIMISATION



(57) Abstract

The present invention relates to a method of designing or optimizing a business. The method involves defining the business as a number of business process objects using object orientated design techniques. Each business process object is designed to provide one or more services in response to an event. This allows the interactions between the business process objects which are required for the business to operate to be modeled. Each business process object and interaction can then be examined in more detail using object orientated techniques.

BEST AVAILABLE COPY

FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania	ES	Spain	LS	Lesotho	SI	Slovenia
AM	Armenia	FI	Finland	LT	Lithuania	SK	Slovakia
AT	Austria	FR	France	LU	Luxembourg	SN	Senegal
AU	Australia	GA	Gabon	LV	Latvia	SZ	Swaziland
AZ	Azerbaijan	GB	United Kingdom	MC	Monaco	TD	Chad
BA	Bosnia and Herzegovina	GE	Georgia	MD	Republic of Moldova	TG	Togo
BB	Barbados	GH	Ghana	MG	Madagascar	TJ	Tajikistan
BE	Belgium	GN	Guinea	MK	The former Yugoslav Republic of Macedonia	TM	Turkmenistan
BF	Burkina Faso	GR	Greece			TR	Turkey
BG	Bulgaria	HU	Hungary	ML	Mali	TT	Trinidad and Tobago
BJ	Benin	IE	Ireland	MN	Mongolia	UA	Ukraine
BR	Brazil	IL	Israel	MR	Mauritania	UG	Uganda
BY	Belarus	IS	Iceland	MW	Malawi	US	United States of America
CA	Canada	IT	Italy	MX	Mexico	UZ	Uzbekistan
CF	Central African Republic	JP	Japan	NE	Niger	VN	Viet Nam
CG	Congo	KE	Kenya	NL	Netherlands	YU	Yugoslavia
CH	Switzerland	KG	Kyrgyzstan	NO	Norway	ZW	Zimbabwe
CI	Côte d'Ivoire	KP	Democratic People's Republic of Korea	NZ	New Zealand		
CM	Cameroon			PL	Poland		
CN	China	KR	Republic of Korea	PT	Portugal		
CU	Cuba	KZ	Kazakstan	RO	Romania		
CZ	Czech Republic	LC	Saint Lucia	RU	Russian Federation		
DE	Germany	LI	Liechtenstein	SD	Sudan		
DK	Denmark	LK	Sri Lanka	SE	Sweden		
EE	Estonia	LR	Liberia	SG	Singapore		

BUSINESS OPTIMISATION

The present invention relates to a method of designing or optimizing a business.

Conventionally, business organisation has been carried out on a manual basis with the result that there is often duplication of tasks within different areas of the business while improvements in business processes which are applied in one area are not applied to other areas because there is no recognition that they are equally applicable. Worse still, they may not be applicable because the business has grown without any consistency or uniformity.

In accordance with a first aspect of the present invention, we provide a method of designing or optimizing a business, the method comprising defining the business as a number of business process objects and their interactions using object orientated design techniques, each business process object being designed to provide one or more services in response to an event.

Accordingly, we provide a method which utilises object orientated techniques to design or modify a business. The invention operates by defining the business as a number of interacting business process objects. The business process objects can then be modelled from scratch or from predetermined business process objects using object orientated design techniques.

Object orientated techniques involve considering systems in general terms to allow similarities between apparently different processes and systems to be established. Such techniques are currently used in software design by considering the relationship between different problems and solutions. Once a relationship is determined, this can be used for adapting software for use in a number of different systems.

This technique therefore allows improvements to be propagated between business process objects so as to improve the overall efficiency of the business as well as

allowing the business to be defined in a manner which allows easy implementation and modification as required. General object orientated methods are described in "Object Orientated Methods A Foundation" by James Martin and James J Odell.

Furthermore, the alteration of the operation of one business process object can cause subsequent alteration of the operation of other business objects within the system, thereby allowing alterations to easily propagate through the business ensuring efficient adaptation of the business to different circumstances.

The method typically involves defining the business as a number of interacting statecharts, each statechart representing a number of states of business processes or sub-processes.

The method preferably comprises defining the business as a number of statecharts, each statechart representing a number of states of the business, the states being interconnected by events and corresponding event response actions, each state corresponding to a respective business process object and each event response action corresponding to a service provided by the respective business process object in response to an event.

The term statechart as used in the invention refers to any definition of a number of interacting states and is not limited to graphical representation. This term should therefore be used interchangeably with the term statemachine. The statechart diagrams are merely graphical representations of the underlying information defined by the statecharts themselves.

Details of how the predetermined business process objects operate are defined in statecharts. Each statechart sets out how different states of a business process object interact to allow the business process object to function. This therefore allows the services which the business process object defines to be implemented more readily.

The method normally requires the determination of service implementations of at least some of the event response actions, the service implementation defining the procedures required for implementing the respective event response action. This allows various aspects of the operation of the business process objects to be defined relatively easily. In particular, the implementation of a given business process object may vary depending on the circumstances in which it is used (known as a polymorphism). However, this can readily accounted for simply by using appropriate service implementations and statecharts depending on the circumstances in which the business process object is used.

Typically the method further comprises defining at least some of the business process objects as respective statecharts so as to generate a hierarchical structure of statecharts. By defining a hierarchical structure, it is possible to identify any services which must be adapted, with greater precision. Furthermore, it will be appreciated that in many circumstances the higher level statecharts themselves do not require any modification per se, although the event response actions of the states may turn out to be incorrect.

In this case the original statecharts are satisfactory and do not require modification. However, the step of analysing the sub-level statechart is still useful as it may allow the event response action to be corrected at a more detailed level of design. Furthermore, the business can be modified on different levels without effecting the operation of other levels of the business. This is particularly advantageous if it is required, for example, to modify the way in which one particular business process object operates when this should not affect the operation of any other business process object.

As will be appreciated, the provision of the hierarchical structure therefore allows different aspects of the business to be modified and optimized as required

with the effects of this being propagated through the business without necessarily requiring modification of the business model as a whole.

Typically the method of determining a statechart of a business process object comprises the steps of:

a. comparing the business process object to a number of predetermined business process objects;

b. determining if one of the predetermined business process objects can be related to the business process object by an object-oriented process; and, either:

i. using a predetermined statechart of a predetermined business process object that is related to the business process object; or,

ii. generating a new statechart.

This allows statecharts which have already been determined for alternative business process objects to be readily utilized when designing or modifying a business. It is also possible to generate the statechart from a consideration of the services and capabilities which the business process object must achieve. However, by modifying existing statecharts, any improvements to the statechart can then be incorporated back into the original predetermined business process object. This not only results in modifications being propagated throughout the business design thereby allowing enhanced operation of a greater proportion of the business but also this allows modification of the predetermined database so that improvements can also be automatically implemented within other businesses.

The predetermined business process object can be a generalization or a specialization of the business process object. Alternatively, identical business process objects may be used.

The method of generating a new statechart usually involves:

a. determining events to which the business process object must respond;

b. dividing the events into sets of related events, each set of related events being associated with a respective state of the business;

c. for each event, defining one or more event response actions for providing the required services.

This advantageously ensures that each state is associated with a set of related events to ensure that states are independent of each other. However, states are then able to interact with each other should this become necessary.

The method generally further comprises incorporating the generated statechart and the associated business process objects into the model of the business. As this occurs, it may become readily apparent that there are inconsistencies in the current business model in which case the model can be adapted as required.

The method of using the predetermined statechart comprises:

a. comparing the services provided by the predetermined business process object with the services to be provided by the business process object; and,

b. if necessary, modifying the selected statechart by performing at least one of the following steps:

i. adding in one or more new states, and interconnecting the new states to existing states using appropriate event response actions so as to provide the required services; or,

ii. modifying the event response actions of existing states to represent changes in the services to be provided by the business process object.

This therefore provides techniques for modifying the statecharts of the predetermined business process objects if they do not accurately model the business process object under consideration. A further alteration that is possible

is the modification of the way in which states interact, even if the states themselves do not need to be altered.

The step of modifying the event response actions of an existing state can comprise modifying the service implementation of the event response action such that the state provides the required services. However, the alteration may alternatively require the modification of deeper level statecharts.

In this case, if the service implementation of the event response action cannot be modified, the method further comprises:

(1) considering the business process object of the state to be modified;

(2) determining a second level statechart of the business process object of the state to be modified;

(3) determining the states of the second level statechart diagram which are to be modified;

(4) for any states for which the service implementation cannot be altered to provide the required services, repeating steps (1), (2) and (3) to generate the nth level statechart diagram for which the service implementation of the state can be altered to provide the required services; and,

(5) modifying the service implementation of the event response actions of the states of the nth level statechart such that the modified statechart represents the services provided by the respective business process object.

By performing this repeated analysis of sub-levels of statecharts, it is possible to identify any services which must be adapted, with greater precision. Furthermore, it will be appreciated that in many circumstances the statecharts themselves do not require any modification but rather the event response action of the statechart may turn out to be a polymorphic operation of the event response action of the statechart of a predetermined business process object. In this case the original statecharts are

satisfactory and do not require modification. However, the step of analysing the sub-level statechart is still useful as it allows the event response action which is polymorphic between the statecharts to be determined.

Typically an nth level statechart diagram is determined from a number of predetermined statechart diagrams in accordance with the n-1th statechart. Again however the state chart may be determined by analysing the services and capabilities of the business process object and of the n-1th statechart.

The method preferably further comprises:

a. considering the modified statechart and the predetermined statechart;

b. considering the effect of inheritance on the operation of the business process object as defined by the modified statechart and the predetermined business process object; and,

c. adapting the relationship between the business process object and the predetermined business process object as required.

Thus, for example, it may be decided that using a specialisation of the predetermined business process object is unsuitable due to the fact that any changes made to the predetermined business process object during operation of the business would then be inherited by the business process object. In this case it may be better to produce a more generalised predetermined business process object and then specialise that to form the business process object which is desired. Accordingly, constant consideration of the interaction between the statecharts and the business process objects can lead to revisions in the business model which then leads to further improvements in the operation of the business. This technique also has the advantages that alterations can be back propagated to the database of predetermined business process objects thereby allowing improvement of other businesses as well as the current business under consideration.

The service implementation may define one or more algorithms configured to perform at least a portion of the respective event response action. Alternatively, the service implementation defines one or more sets of instructions indicating the tasks to be performed in order to perform at least a portion of the respective event response action. Alternatively, as will be appreciated by a person skilled in the art, the service implementation may constitute a combination of instruction procedures to personnel of the business, as well as algorithms which are operated by processing systems to generate certain data. Alternatively the service implementation may be out sourced to a different company depending on the circumstances in which it occurs. In this case, the service implementation would simply be to obtain information, and/or goods or services from an external source.

It will also be realised that the service implementations may be defined from scratch by considering the nature of the service and considering the best way in which this could be provided in accordance with the current business object under consideration. Alternatively however, the method preferably further comprises determining the service implementation diagrams for a given event response action of a state by:

a. comparing the business process object associated with the state to a predetermined business process object;

b. determining a service implementation for an event response action of the state of the predetermined business process object; and,

c. modifying the service implementation to allow the required event response action to be implemented.

This has the advantage that it allows service implementation procedures which are known to work to be implemented within the business thereby ensuring that the service will be provided as requested.

The method can further comprise determining the core capabilities of a state of the business, the core capabilities representing the business process objects and their respective interactions required for the state to be implemented. This is a useful conceptual view of the business which allows the technique user to have confidence that the designed system is meaningful and functional.

In this case, the method usually also involves:

a. comparing the core capabilities with a number of predetermined core capabilities for different business states corresponding to the predetermined business process objects; and,

b. modifying the core capabilities and the statecharts in accordance with the predetermined core capabilities.

The obtained core capabilities diagrams can then be compared with the statechart to check that the business defined by the statechart is a reasonable business model.

The method typically involves generating statechart diagrams representing the states, the event response actions and the event for a given statechart. Core capabilities diagrams showing the business process objects and their respective interactions which represent the core capabilities of any state of the business are also usually generated. The production of the diagrams allows the business manager to visualize how the generated business model will function in reality. In particular, it allows the business manager to examine statecharts and core capability diagrams at different levels within the business to obtain an idea of how the business process objects, and the various states of the business are interacting.

The object orientated processes usually include the use of inheritance and polymorphism.

The method may also further involve, presenting business data by associating business data with the statecharts and/or the core capabilities. This allows the data to be viewed on the appropriate statechart and/or core capabilities

representation making it easier for the business manager to determine the relevance of the data to the business model.

It will be realized that the use of the above described techniques in combination, and notably the use of inheritance, statecharts and polymorphism, provides significant benefits over previous business modeling techniques. This is because of the inherent interaction between the techniques which allows the business to be more easily modeled.

A further problem that has arisen in previous consideration of businesses is the calculation of the risk associated with the business failing to operate correctly. The risk of a business failing is a value required by law for many industries but this value is currently calculated on the basis of the size of the business and the income of the business. The value is actually determined using the equation:

$$Risk = \alpha A + \beta B \qquad (1)$$

where: α = size coefficient
A = size of the business
β = income coefficient
B = income of the business

Although the value of the coefficients can be altered, this technique has the particular disadvantage that large firms with a high income tend to have a high risk value even if the company itself is not particularly risky.

In accordance with a second aspect of the present invention, we provide a method of risk assessment within a business, the method comprising the steps of:

a. defining the business as a number of business process objects and their associated interactions using object orientated design techniques, each business process object being designed to provide one or more services in response to an event; and,

b. determining the risk of at least some of the business process objects not providing a service in response to an event.

The present invention overcomes this by determining a separate risk for at least some of the business process objects which define the business. This allows the risk to be determined on a far more accurate basis by basing the risk on the possibility of a portion of the business failing to operate correctly rather than on the basis of an abstract equation which does not reflect the actual operation of the business in any way.

In the situation in which the business is designed in accordance with the first aspect of the invention the method of determining the risk preferably further comprises:

a. assigning a risk value to each event response action of each state of the business process object, the risk value being representative of the risk of the event response action not being performed;

b. defining a relationship between the event response actions of the states; and,

c. determining a risk value representing the risk of a business process object not providing a service, the risk value being determined on the basis of the risk values associated with each event response action and the relationship therebetween.

The risks can therefore be related to specific events within the business and by combining the risk values for each event the risk value of a business process object not performing a desired service can be determined. This allows the model to take into account the fact that failure to provide one event response action could lead to failure of other related event response actions. Alternatively, the risk value could be assigned directly to the business process object.

The method of assigning a risk value to a given event response action comprises determining the likelihood that the respective service implementation cannot be completed.

Typically the method further comprises assessing the risk of a failure in a business by:

a. determining the risk of each business process object not providing a service in response to an event, for all events and all business process objects;

b. determining a relationship between the business process objects; and,

c. determining a risk based on the risk associated with each business process object and the relationship therebetween.

This allows a single value to be determined for the entire business. Whilst this value does not relate to any one event in particular, the components of the event can be determined by consideration of the business process object and statechart structure of the business. The single values have a useful implication of the overall function of the business. In particular, should this value increase significantly this would suggest that some area of the business is having trouble implementing the services currently allocated to it. This would therefore allow the business to be redesigned with a lower risk state, thereby improving the efficiency and operation of the business.

In accordance with a third aspect of the present invention, we provide a method of designing or optimizing an Internet based business system, the business system being formed from a number of service providers, the method comprising defining the business system as a number of business process objects and their interactions using object orientated design techniques, each business process object representing a respective service provider and being designed to provide one or more services in response to an event.

The present invention therefore provides a malleable business design which allows the designed business system to

be highly flexible so that the system provides a highly reconfigurable set-up where services are out-sourced, co-sourced, in-sourced and/or shared as appropriate. This reflects the dynamics of the ever changing business relationships which typically occur for Internet based businesses. This system therefore allows alterations in the industry to be readily integrated in to the business model, whilst improvements in the business model can readily be propagated into the working business system.

The method preferably uses a number of business process objects to represent service consumers which interact with the business process objects which represent the service providers to receive desired services. However, customers need not be represented in this way and can instead, for example, be represented as events for the request of services within the business model.

Typically the method is implemented using the methods of the first and second aspects of the invention. Thus the risk assessment of the second aspect of the invention can also be applied to the Internet based business model.

Typically at least some of the service implementations are implemented using Enterprise Java Beans. This allows the business model to define the service implementations in sufficient detail for many services to be automated in accordance with the operation set out in the model.

An example of processes according to the present invention will now be described with reference to the accompanying drawings, in which:-

Appendix Drawings

Figure A is a schematic representation of a business process object;

Figure B is a schematic diagram of a request for approval of a loan from a loan application business process object;

Figure C is a schematic diagram of a non-core capability; and,

Figure D is a schematic diagram of the inheritance of services from a first business process object by a second business process object which is a specialisation of the first.

Description Drawings

Figure 1 is a schematic representation of three complementary views of object orientated analysis;

Figure 2A is a schematic diagram of a service request to a customer business process object;

Figure 2B is a schematic diagram of a customer business process object;

Figure 2C is a schematic diagram of the delegation of business responsibilities between business process objects;

Figure 2D is a schematic diagram representing the distribution of business responsibilities to people and systems within a business process object;

Figure 2E is a schematic diagram of a business process object which has been modified to encapsulate people and system capabilities;

Figure 2F is schematic diagram of a modification of the business process objection in Figure 2D in which additional automation has been included;

Figure 2G is a schematic diagram of the provision of mortgage services using a mortgage account and customer business process objects;

Figure 3A is a schematic diagram of a business;

Figure 3B is a schematic diagram of a statechart of the business shown in Figure 3A;

Figure 3C is an example of the business process object representation of the business shown in Figure 3A;

Figure 3D is an example of a modified version of the business process object representation of Figure 3C;

Figure 4A is a service implementation diagram for the "open an account" business service;

Figure 4B is a simplified statechart for the account business process object of Figure 4A;

Figure 4C is an example of a graphical user interface representation of the states of Figure 4B;

Figure 4D is a schematic diagram of the aggregate responsibility distribution for "opening an account";

Figure 4E is an example of a graphical user interface of states modified in accordance with Figure 4D;

Figure 4F is a schematic diagram representing the organization reverse engineering of a business;

Figure 4G is a schematic diagram representing the process design of a business;

Figure 4H is a schematic diagram representing the creation of a business design;

Figure 5 shows an object hierarchy diagram showing the proposed relationships between the FX/IR trade and credit derivatives;

Figure 6 shows a modified version of the object hierarchy diagram of Figure 5 including the modifications that are required to the credit derivative and FX/IR trade designs;

Figure 7 shows the statechart diagram for the FX/IR trade administration business process object;

Figure 8 is an object hierarchy diagram showing a modified relationship between the FX/IR trade object and a credit derivative object;

Figure 9 is a modified version of the statechart of Figure 7;

Figure 10 is a second level statechart sub-diagram of the trade management state shown in Figure 9;

Figure 11 is a third level statechart sub-diagram of the management cash flows state shown in Figure 10;

Figure 12 shows a modified version of the object hierarchy diagram shown in Figure 8;

Figure 13 shows a list representing the relationship between the statechart diagrams of different levels;

Figure 14 is a poor capabilities diagram floor trade administration in the credit derivatives business process object;

Figure 15 is a second level core capabilities sub-diagram showing the core capabilities of the trade management core capability of Figure 14;

Figure 16 is a third level core capabilities diagram showing the core capabilities of the cash flow management core capability of Figure 15;

Figure 17 is a fourth level core capabilities sub-diagram of a management subsequent fixings core capability of Figure 16;

Figure 18 is a service implementation diagram for the surface rate fix of Figure 17;

Figure 19 shows an example of the firmament framework of the present invention;

Figure 20 shows the meta-model of the method of the present invention;

Figure 21A is a representation of the core capabilities of Risk Management within a client bank;

Figure 21B is a representation of an object hierarchy associated with the core capabilities of Figure 21A;

Figure 22 is a modified version of the meta-model of Figure 20, modified to provide operational risk assessment and management reporting; and,

Figure 23 is an example of the core capabilities of Risk Management and Capital Allocation.

The main concepts which are used to design or modify a business using the present invention are set out in Appendix A, below. The main forms of diagram used by the invention are briefly explained in Appendix B.

Firmament Framework

The approach of the present invention is based on the identification and the design of business process objects (business design). This then needs to be followed by a well-planned implementation and roll-out of business process objects (change management). This process which is known as the Firmament Framework is an iterative procedure in that once a section of the business has been analyzed in a given way, the resulting model is compared to previously determined models so that further adaptions can be made.

An outline of this process is shown in Figure 1 and this shows how the three main techniques which are used, namely the business process object modeling, the behavior modeling using statecharts and the implementations are interrelated. Accordingly, the modification of the model derived by one technique may effect the model determined by another technique. Once this effect has been included in the model, the first model may then require further modification.

The main concepts of the invention will now be described, followed by examples of the operation of the invention.

Business Design

Only carefully designed business process objects should be used when attempting to build an effective business. In general, good designs provide a stable base for future improvements, whereas poor designs are the root of a rapid deterioration.

A very important concept is that of object designs which serve as useful blueprints for construction of business process objects. New objects can be built rapidly from the design whilst preserving the consistency at all times. When business process objects need to be changed the complexity of that exercise may be greatly reduced by the fact that many of them share the same design.

Clearly defined object designs and their enactments, the business process objects, are major assets of an effective business.

Object designs (and therefore the business process objects based on these designs) encapsulate system capabilities and people competencies and the interactions of the two.

An object design is a complete and unique (encapsulated) business asset which defines a set of closely related business capabilities or services.

For example, the Mortgage Account object design shown in Figure D (and discussed in the Appendix A) encompasses, all the business rules and system supports needed to provide a mortgage as well as all the people competencies required for example to authorize a payment break. Competencies are carried out by roles, such as a Mortgage Account Manager.

When designing effective businesses, it is necessary to avoid having People, Team, Person, Account Manager as identities separate from business process objects. Instead peoples roles should be defined in object designs and contained in enactments thereof, i.e. in business process objects.

Similarly, the system details (databases, networks, computers...) should be avoided at early stages of business design as the associated technology is not stable and changes almost overnight.

The most striking example is that of the Customer object design which is effectively the group of related services which allow a business to interact with customers. This business asset does not represent the person on the street, or a client from an external business but, instead, encapsulates all capabilities available to the organization to deal with the person on the street, or with the business client.

It should be noted that all in this context is not extended to advocate 'big' objects. Business process objects may, by their own Business nature, have numerous

responsibilities but more often than not many of these can be delegated to other, better equipped business objects.

A Customer object design will include much more than customer data on the database and the services to display or change this data. It will include the organization's ability to communicate with the customer, (e.g. using telephone equipment and the people well trained in telephone manners), as well as the ability to deal with customer requirements. It should be noted however that a Customer business process object should not be considered solely as the "Customer Liaison Department". In an effective organization there will be very few centralized departments.

A Customer business process object will deal with one or more customers and, typically, there will be more than one Customer object in the business.

In order to be provided with a service from a business process object, it is necessary to request it.

Figure 2A is a schematic diagram of a service request to a Customer business process object, the service being to"inform the Customer".

The assumption is that the Customer object design defines a business service of the same name. This definition generally includes performance requirements, such a minimum return on an investment or a minimum level of customer satisfaction. The customer business process object shown in Figure 2B includes an indication of the business services which are defined by the customer object design. This is important as it allows business designers to maximize the re-use of assets.

In use, the business process objects co-operate in order to meet their individual responsibilities. Thus as shown in Figure 2C, a Mortgage Account business process object delegates the responsibility of performing a service (in this case informing the customer) to an alternative object (in this case the customer object) which is able to provide the service. The definitions accompanying the design should also show the intended delegation of performance targets.

Change Management

With people and systems aspects firmly encapsulated in object designs, once the business responsibilities become clear it is possible to design people roles and the supporting system.

First of all, when there is no obvious other business process object to delegate to, the responsible business process object distributes this responsibility internally, to the encapsulated people competencies and system capabilities.

Thus if there is no other business process object which obviously provides a given service, then the business process object will be adapted to provide the service itself.

An example of this is shown in Figure 2D. In this case the service to be provided is informing the customer, which is achieved by displaying the customer details and then telephoning the customer. As no other business process objects provides these services, then the customer business process object is adapted as required. In this particular adaptation the customer object uses system capabilities (denoted by s_) to display the customer details and personnel competencies (denoted by p_) to telephone the customer.

It will be realized that the capabilities to display Customer details and to telephone the Customer are internal (private) to the Customer business process object and are not normally accessed from other business objects. However, Customer will not become a bottleneck in a well-designed organization. Business process objects are usually replicated and distributed, i.e. there will be a number of Customer objects providing the necessary services across the business.

A modified version of the Customer object of Figure 2B is shown in Figure 2E. In this case, the Customer object has been modified to include an indication of the display and telephone business services which are defined by the customer object design.

The separation of business designs from the implementation details means that the responsibility

distribution can be continuously improved without affecting the core business. Thus in the example shown in Figure, 2F, the service capabilities of the customer object are improved such that the system not only displays the customer details but also automatically dials the Customer's telephone number (denoted by s_dial) thereby reducing the work required by the personnel competencies.

When the business design and the responsibility distribution are stabilized and agreed, it is possible to start the role design and system design.

After consultations with 'Human Resources', the role design could be as shown in Table 1.

TABLE 1

Role Design	**Roles** (shading indicates changes since the previous design)			
Competencies	Help-desk Advisor	Customer Manager	Mortgage Account Manager	Loan Officer
approve the Loan	N.A.	existing	new	existing
talk to the Customer	modified	modified	N.A.	N.A.
authorize overdraft	N.A.	new	N.A.	N.A.

Taking this into account, a first draft of a system design for the provision of mortgage services would be as shown in Figure 2G.

Outline Example

A general example of the method of the invention will now be discussed. It will be appreciated that whilst the term business is used throughout the description, the present invention could be applied either to an entire business, or to a discrete portion of the business, such as a subsidiary company only.

The first stage is to define how the business interacts with the external world. This can typically be achieved by considering and providing a high level definition of the

business capabilities which need to be developed.

For each of these capabilities, which are implemented as the provision of services, a short but precise statement of what is required, but not how it might be implemented, can be produced. Thus, for example, in the case of opening a new bank account the statement could be as follows:

"A new account will be created subject to checking the customer and checking the application. The customer will be informed about the outcome."

Once all the capabilities have been defined, the business can be considered as a single business process object which interacts with the external world by providing the defined services in response to events which define a request for such a service.

Figure 3A shows an example in which the business is a bank. In this case, the bank business process object 1 must be able to respond to events which are indicated generally by the arrows 2, and examples of which include a request to open an account, and a request to provide security for cash deliveries/collection. The responses are in the form of the provision of services which are indicated by the arrows 3 such as the opening of an account and the provision of security.

Once the business has been defined in these terms, it is then possible to derive a statechart representing the operation of the business process object. The statechart may be obtained from a database of predetermined statecharts, or alternatively may be generated by consideration of the events and services associated with the business process object, as will be explained in more detail below.

In some circumstances the previous development of businesses will result in the generation of business process objects which are suitable for modeling the business project object of the current example. In this case, the statechart can be determined from the statechart of the previously

designed business process object.

In order to determine whether a predetermined business process object is suitable for use as the business process object under consideration, the two objects, the events to which the objects respond and the services provided are compared. In general the predetermined business process object is deemed to be suitable if it:

- provides all required services, or
- can be extended without distorting the core behavior of the object, or
- a new object design can be derived by specializing the existing one, or
- a new object design can be derived by generalizing the existing one.

The statechart of the business process object therefore may require modification which will be explained in more detail below.

In order to generate a new statechart, it is necessary to divide the business process object into a number of states. This is achieved by separating the events associated with the business process object into a number of related groups, with each group being associated with a respective state of the business.

Thus, in the example of the business being a bank, the various transactions dealing with customer accounts could form one group of related events, whereas the provision of security to the bank will form a separate set of events. Accordingly, in this case, the statechart would be of the form shown in Figure 3B, in which an account state 4 and a security state 5 are provided.

Once the states have been defined, event response actions can also be defined for each state as shown by the arrows 6. The event response actions represent the action that is needed by the respective state in order to ensure that the required service is provided.

Each state in this statechart itself corresponds to a respective business process object, with the event response

actions representing the services provided by the respective business process object.

Each business process object can therefore be considered in more detail. Again, this may be achieved by comparing the business process object to a number of predetermined business process objects, or by generating a statechart for the business process object by considering the events to which the business process object must react and the services which the business process object will provide.

It will be appreciated that this process can then be repeated to form a hierarchical structure of statecharts which define the operation of the business. Each statechart therefore provides details of how a respective business process object will operate to provide the required services. Furthermore, each state of the statechart itself represents a business process object such that the statechart itself represents how a number of more narrowly defined business process objects must interact to enable a more broadly defined business process object to be implemented.

The procedure of the present invention therefore enables further and further levels of statecharts to be generated with each successive statechart providing further details of a more narrowly defined area of the business than the previous statechart. This process is generally continued until the states of the statechart define event response actions in sufficient detail to allow them to be implemented.

The implementation of event response actions is achieved by defining service implementations which are a series of procedures required to ensure the desired event response action is performed following the occurrence of the respective trigger event. The event response actions therefore represent the procedures that must be carried out to enable the business process object to implement the service which corresponds to the respective event response action.

The service implementations may therefore comprise a wide range of different procedures from a series of

instructions to personnel within the firm, to a computer program designed to automatically respond to a request for information.

Furthermore, the service implementations can be generated at any stage to aid the modification and creation of other portions of the business model. Thus, for example, the process of defining a service implementation for an event response action may highlight that the state which is generating the event response action needs to be defined in more detail.

Again, where the event response action is simply a copy of a service which is already implemented by another business process object, then the service implementation can be copied from the service implementation of the other business process object, making any modifications that are required.

If however no previous similar service has been implemented, then it is necessary to determine the service implementation diagrams from a consideration of how the service is to be provided.

Returning to the example of the statechart shown in Figure 3B, the statechart includes an account state which in turn corresponds to an account business process object which is configured to handle account enquiries.

This situation is shown in Figure 3C which represents the business process objects, namely an account object 8 and a security object 9, so far defined. At this stage, these represent the core capabilities of the bank 1.

In this example, the account business process object can be defined in more detail by considering previously defined business process objects. Assuming that the database contains only the object designs which are mentioned in the appendix, then the most similar business process object is the Mortgage Account object design which handles mortgage accounts.

As mortgages are a specialized type of account, the account business process object can be considered to be a generalization of the Mortgage Account object design.

However, in contrast to the Mortgage object design, the Account object design requires a further capability, namely the capability to process applications. Accordingly, the object design, when generalized is modified to include a processing applications capability.

This can be achieved by direct consideration of, and modification of the statechart of the Mortgage Account business process object.

However, in order to modify the statechart, the procedures that are required to process applications must first be determined. In this example, this is achieved by determining the service implementation that is required for the account business process object to process applications. In this case, the service implementation is produced in the form of a diagram showing the interactions that are required between the business process object and the customer and application business process objects, a suitable example of which is shown in Figure 4A.

In this diagram the instructions are developed in a clockwise manner. The sequence of events is therefore as follows:

1) Receive an "open" request at the Account Object.
2) The Account Object requests a check of the Customer via the Customer Object.
3) The Account Object then checks the application via the Application Object.
4) After receiving confirmation that the application is acceptable, the Account Object opens the account.
5) The Account Object instructs the Customer Object to inform the Customer that the account is open.

The documentation work arising from this diagram involves:

- Account object design
- Application object design
- Definition of the service to check the Customer

Accordingly, it is realized from this that it is necessary to modify the model of the bank to include an application business process object 10 and a customer business process object 11, as shown in Figure 3D. Unlike the security business process object 9 which does not interact with the account business process object 8, the application and customer business process objects 10,11 interact with the account business process object 8, as shown by the arrows 12.

It will be realized that this information could also have been obtained by considering a core capabilities diagram of a bank business process object. This would have shown that the bank business process object would require customer and application business process object in addition to security and account business process object. Accordingly, several different approaches can be used in conjunction or independently to obtain the same result.

Furthermore, the service implementation diagram of Figure 4A highlights that the requirement to process additional accounts results in the account business process object having a number of states not inherently present in the mortgage account business process object. Accordingly, the statechart of the account business process object would require the addition of the states shown in Figure 4B.

This allows the operation of the Account object to be monitored and modified as required. (It should be noted that Figure 4B does not show events which stimulate state transitions. Also, beware that some events do not cause state changes.)

An example of how this statechart information may be represented on a graphical user interface (GUI) is shown in Figure 4C. Even if the GUI subsequently needs to change, it is essential to maintain traceability with the statechart.

Consideration of the statechart representation on the GUI also gives rise to at least 3 new 'use cases':

- Suspending an Account
- Re-opening an Account

- Closing an Account

There are also a number of questions, such as which services should be provided for suspended accounts. As a result of this consideration of the service implementation diagrams and the statecharts, further modifications can therefore be made.

The next step is to distribute responsibilities to people and systems for all 'leaf' business services, i.e. those which are not further elaborated on other diagrams. This can be done by designing an aggregate responsibility distribution for the whole service implementation diagram as shown in Figure 4D.

The accompanying definitions should also show the intended distribution of performance targets. The GUI development might include the modifications shown in Figure 4E, based on the people responsibilities identified in Figure 4D.

To re-iterate, even if the GUI subsequently needs to change, it is essential to maintain traceability with the design.

When the responsibility distribution is fully agreed between the business analysts, the 'human resources' and the systems specialists, the work can commence on the detailed design and implementation of people competencies and system support.

Ultimately, the resulting Clients' business process objects will be not only compatible with the 'structure' of their business - they will effectively become the business ('Object Orientated business' or `OO' Business for short). The difference between an OO business and conventional businesses is best illustrated by describing the types of services the invention can provide:

(i) organization reverse-engineering - where a 'road-map' of the existing business functions is created, as shown in Figure 4F;

(ii) process design - cutting right across functional barriers thus making the business process

driven, as shown in Figure 4G;

(iii) business design - creating a flexible-to-change business based on re-usable building blocks (business process objects) thus creating an 'OO business', as shown in Figure 4H.

Implementation

The present invention will generally be implemented on a suitable processing system. The advantage of using a processing system is that the data representing the various hierarchical levels of the business model can be linked so that the various levels can be easily viewed.

Thus, for example, each core capabilities diagram which shows the business objects and their respective interactions would also be linked to statechart diagrams so that selecting a given business process object results in the statechart of that business process object being displayed. Similarly, once a statechart is displayed, any state within that statechart can be selected and the appropriate business process object and its associated core capabilities can be displayed.

In order to achieve such a system, it is therefore necessary to have the statechart, the core capabilities and the service implementations of the business set out in suitable look-up tables in a memory. Each look-up table can then include links to other look-up tables as appropriate.

In particular, this has the advantage that when predetermined business process objects are used, the tables can be stored in a database and the information contained therein can easily be incorporated into the current model by including a reference to a table within the database.

By providing the suitably adapted processing system, this allows the statechart, the core capabilities and the service implementations to be presented as diagrams which can therefore be readily understood.

Furthermore, by using an appropriate processing system

any design changes implemented in one section of the business model can automatically be integrated to other sections of the business model, as appropriate. Furthermore, any changes at a more detailed level will automatically propagate through to higher level definition of the business.

Figure 20 shows a meta-model which sets out the steps involved in providing software implementation of the method of the present invention. This highlights the steps needed to perform the method so as to define and optimise a business.

Design Focusing

From the above, it is apparent that as the business is developed an iterative process occurs in which the consideration of various aspects of the business results in the modification of other aspects. If the business were considered as a whole, this would be prohibitive to the successful modification of the business. However, by studying smaller business process objects which relate to a specific portion of the business, this allows the business to developed in a modular manner, with each part of the business being defined in sufficient detail to allow it to be implemented.

A particular advantage of this is that when the business model is considered, it is possible to examine the business model at different levels of detail which are appropriate to the observer. This therefore allows people to focus on the business at a level which is relevant to them.

Thus, in the present example, an employee in the tax section in a sub-branch in one country will not have any interest in how the business as a whole operates. In contrast to this, the director of the company will be very interested in how the business operates but will not necessarily be interested in how tax is calculated in a sub-branch in a particular country.

Polymorphism

A further advantage of the technique of the present invention arises from the object orientated feature of polymorphism. Polymorphism means that similar business process objects may have to respond to similar events so as to provide similar responses. However the manner in which the business process object is implemented may vary significantly depending on the circumstances in which it is used.

Thus, in the present example of a bank, typically there will be provided a business process object for calculating tax. However, in the situation of an International corporation, the actual tax calculations that have to be performed will vary from country to county. Thus, for example, the specific calculation of tax in the UK will be different to that in the US and again to that in Japan.

However, the general operation of the tax business process object will be standard in that the tax business process object will respond to a calculate tax event to calculate the required tax. Thus, as the tax business process object will function in the same manner regardless of the country or size of bank branch in which it is implemented, it can be used throughout the world.

However, the specific calculation of the tax, will vary depending on the branch location. Accordingly, the specific service implementation of the tax business process object will be different for branches in different countries to reflect the differences in the calculations required to determine the tax payable. So although the tax business process object reacts to the same event and provides the same end response, the actual implementation is vastly different.

In this case, the statechart of the tax business process object itself will also be substantially similar throughout the globe, as the basic events in the procedure of tax calculation are substantially the same. Thus the general steps are to receive tax request events, to process the tax due and provide an indication of the tax due.

However, when this statechart is considered in more depth, the specific implementation will vary between each country. As the implementation is itself defined by more detailed statecharts further down the hierarchy, these statecharts may well include variations between the different countries.

Thus for example, for the calculation of the tax in Japan it may require only a single person working to calculate the tax on a set of given transactions. In contrast to this, in alternative countries such as the UK an entire team of people using a complicated computer software system will be required.

These differences will be reflected with detailed statecharts and service implementations for the tax business process object.

Thus at a higher level, which for example would be considered by a company director, the tax business process object appears identical, even though the implementation is different. Accordingly the tax business process objects are a polymorphism.

A second example of a polymorphic relationship is shown in Figures 21A and 21B. Figure 21A represents the core capabilities required by a risk management department in a bank and this is intended to provide certain uniformity of implementation (not only for regulatory reasons) across the Bank.

Figure 21B is an object hierarchy diagram which shows that the uniformity can be preserved even when extending the generic capabilities of Risk Management for different business lines, e.g. Risk Management in Debt and Capital Markets (DCM) has, additionally, the capability called "spread risk". Furthermore, and equally importantly, Risk management in Fixed Income (FI) also has a "spread risk" capability, but the implementation may be different to that in DCM. This is why polymorphism is sometimes referred to as the "customizable uniformity".

An additional advantage of polymorphism is in increasing

productivity i.e. business dynamics. For example, as soon as Risk Management, Risk Management in DCM and Risk Management in FI agree upon the core capabilities required for Risk Management, the latter two can start implementing their spread risk capabilities. Note that DCM and FI do not have to agree how (although it would make sense if they did not use overloaded name for completely unrelated different capabilities) the spread risk capabilities are implemented.

Furthermore, DCM and FI have in common, at least, the statechart defined by the generic Risk Management. Conversely, although they may share even the part of the statechart which deals with "spread risk", when that capability is required DCM will respond by invoking the Sybase/C++ implementation of an algorithm STP34Zx whereas in FI, a dedicated person use a pencil and paper to perform a manual calculation based on departmental method MYOWN46.

Accordingly, by using a polymorphism this allows the business to be defined in a far more efficient manner. In particular, this allows the business to function in the same manner globally at the scale of the tax business process object but in the specific implementation which could be calculated on a country by country basis, there are variations which are required for the tax business process object to function in a suitable manner.

Risk Calculation

An additional application of the present invention is in the calculation of the operational risk of a business.

The present invention calculates an overall risk by determining a separate risk for the provision of each service by a respective business process object. The risks can therefore be evaluated for business process objects at various levels of the hierarchical structure of the business and the results propagated through to higher levels.

In general, it is easiest to start at the most detailed level which looks at the implementation of specific services within the business. Thus, for a group of business objects,

a risk may be assigned to each business process object capability on the basis of the risk of the respective service not being provided in response to the respective event.

By evaluating this for each business process object capability, and considering the interaction between the different business process objects, it is possible to determine from this how failure to perform the given interaction will effect the remainder of the business.

Thus, in a group of closely related business process objects, it may be that failure of one business process object to provide a service may in turn mean that other business process objects will also fail. This fundermental connection between the business process objects is inherent in the model derived by the present invention.

For example, in the case of the core capabilities of the Integrated Risk Management and Capital Allocation structure shown in Figure 23, if Group Finance fails to adequately provide the service "capture trade data in GI" then it may also fail to provide other services it has contracted-by-design, such as:

a. provide financial data for audit (to External Auditor, Tax Advisor)
b. provide financial data (to Regulators)
c. determine how much capital there is (to Calculate ROC and Allocate capital)

The present invention therefore provides a technique which uses a step-by-step analysis of failures of certain portions of the business to allow a risk value to be evaluated and to determine how a failure may effect the overall operation of the business.

In the present invention, as each higher level business process object is defined in terms of a number of interconnected lower level business process objects, the risk of a business process object failing to provide a service can be calculated as a non-linear sum of the risks associated with interactions of the respective lower level business

process objects.

By performing this non-linear sum at each level of the hierarchy, a value representing the overall operational risk of the entire business can be evaluated.

This particular value may not have a meaning in the sense that it does not provide an indication of where the business will go wrong. However, it does provide an indication of the current operating risk of the business. If this is too high, or at some stage during the operation of the business shows an increase, this allows people to see that improvements need to be made to the business.

In a system designed according to the present invention, a high operational risk can be examined in great detail by looking at the different levels of the business model to determine if a weak point exists which contributes significantly to the overall risk.

Thus, for example, the managing director may only look at the overall operation of the business in terms of its core capabilities. From this, the operational risk associated with the business, and with each of the core capabilities can be determined. This may identify to the director that one of the core capabilities has an exceptionally high operational risk.

The director can then request that the person in charge of that particular core capability identify why this risk is so high and then operate to reduce the risk. The head of the indicated section can in turn look at the statechart and/or a more detailed business object representation of the respective core capability. This allows him to determine which respective business process object out of all the business process objects defining the core capability has the highest risk.

By following this procedure through, it allows a business process object deep within the structure of the firm which has an unduly highly risk to be identified, thereby allowing the risk to be addressed so that it can be reduced which in turn improves the overall operating efficiency of

the business.

The present invention therefore allows people to focus on the risk involved in the business at a level which is relevant to them. Thus, an employee in the tax section of the branch of a bank will not have any interest in the risk value for the business as a whole but will be interested in the risk involved in providing the required tax calculations. In contrast to this, the director of the company will be very interested the risk associated with the entire business but will be less interested in the risk involved in specific operations, unless this unduly effects the overall risk value, in which case it becomes more relevant.

However, because the present invention uses software which operates to model all the systems within interrelated charts, each member of the company can use the same model to determine the risk value for the company at a level of interest to them.

In addition to this, the present invention can be adapted to implement a management report system showing possible domino effects arising from a single failure point. In this case, the software used to model the business is adapted to automatically generate the reports for a selected business process object. The reports may be generated on request, or alternatively, the software may be adapted to detect when a failure occurs (using the data linking described below) and then generate a report highlighting the effects of this failure.

In the current example, the Bank's process Calculate ROC and Allocate Capital and external regulators and External Auditor, Tax Advisors are immediate candidates for failure. Examples of suitable reports that could be generated are shown in Tables 2 and 3.

Figure 22 shows a meta-model which sets out the method of the present invention when adapted to provide both risk calculation and management reporting.

Table 2

Source of Risk	Nature of Risk and its Impact[1]	Likelihood of Risk Occurring[2]	Severity of Impact[3]	Mechanisms to Mitigate Risk	Cost and Time-scales of Risk Mitigation	Residual Risk after Mitigation	Contingency Plans	Last Reviewed	Action Required and by Whom / When
Group Finance "capture trade data in GL" : Manual interface from Business Lines to GL in Group Finance	1.unable to provide accurate financial data for audit, which leads to reputational risk arising from qualified audit opinion	H	H	(i) automate the process (ii) improve reconciliation (iii) pre-audit review by an external party	(i) £0.5m in 6 months (ii)£35K pa in 1 month (iii) £60K pa	(i) original risk eliminated (ii)likelihood reduced by 80% (iii) likelihood reduced by 95%	Post-audit review by an external party, costing £150K	3/10/99	Group FD to implement (iii) as a short term measure and (ii) within 3 months; to report back on progress in 2 months
	2. unable to provide accurate financial data to regulators, which leads to poor reputation with regulators, resulting in more frequent visits, higher regulatory capital	H	H	(i) and (ii) as above	(i) and (ii) as above	(i) and (ii) as above	(i) or (ii)		(ii) as above
	3. unable to provide accurate data to Calculate ROC and Allocate Capital, which leads to inaccurate performance reported to shareholders and inadequate capital allocation	H	L	(i) and (ii) as above	(i) and (ii) as above	(i) and (ii) as above	plans for 1. and 2. should deal with 3.		(ii) as above

[1] Reputational, legal, loss, uncaptured risk, environmental, social
[2] H / M / L
[3] H / M / L or, preferably, quantification of risk

SUBSTITUTE SHEET (RULE 26)

Table 2 (Continued)

Source of Risk	Nature of Risk and its Impact[4]	Likelihood of Risk Recurring[5]	Severity of Impact[6]	Mechanisms to Mitigate Risk	Cost and Time-scales of Risk Mitigation	Residual Risk after Mitigation	Contingency Plans	Last Reviewed	Action Required and by Whom / When
At any one time, a number of unsigned ISDA agreements	Unable to enforce collateral and/or netting risk mitigation mechanisms within agreements. The impact[7] is an unwanted and unexpected credit risk to counterparty.	H[8]	£17.6M= unmitigated credit exposure plus add-on for uncertain future events[9], including all the deals we would have done	(i) attempt to secure signed agreement (ii) buy insurance (iii) stop trading with the counterparty (iv) invoke break clauses where and when applicable	(i) under £500 per counterparty (1 week) (ii) anything up to £17.6M, 1 week (iii) opportunity loss, estimated by the Relationship Management to be around £2M pa, with immediate effect. Plus the cost and delay in establishing new relationship (iv) the initial contract specifies the formula	(i) risk eliminated (ii) original risk contained (iii) no new risks can be incurred but original risk remains (iv) original risk is contained	(i) continue attempting to secure signed agreement (ii) negotiate to unwind or re-structure the contract, or attempt to take some form of collateral	15/11/99	Business Line head to attempt (i) and report back in 10 days

[4] Reputational, legal, loss, uncaptured risk, environmental, social
[5] H / M / L
[6] H / M / L or, preferably, quantification of risk
[7] The "least worst" case is where collateral threshold is reached and counterparty have not surrendered collateral.
[8] Unless there are specific circumstances which would make it Low or Medium i.e. the default is High.
[9] For each counterparty that has not signed, the risk arising from not having collateral = MtM - collateral threshold + PFE, and the risk arising from not having a netting arrangement is: gross exposure - net exposure + PFE.

SUBSTITUTE SHEET (RULE 26)

Table 3

Generic Report Type	Example Reports for Board	Example Reports for Group Finance	Example Reports for Limit Management	Directly or indirectly implied changes to the meta-model
Summary of 'outputs' from a process [1,2]	• summary of performance against targets	• capital usage/capital available • budget usage	• summary of approved limits by type and size • key changes to limit type/size since last report	• **for each service request,** measurement of the actual KPI (see SLA, below) plus ad-hoc exception report [3] (or, more likely, a ActiveX "link" to)
Summary of key 'inputs' to a process [4,5]	• summary of internal and external audit • summary of management accounting info, including revisions to forecasts • actual risk v risk appetite, including data on major risks	• data for setting provisions • key asset/liability data • business cash/funding requirements	• summary of requests being processed	• **for each service request,** target KPI plus exception reporting rules [6]
Summary of *internal* (to the process) or *external* events/changes which impact process in a major way		• eg new regulatory capital rules in pipeline which will materially increase capital charge	• eg change in enforceability in a jurisdiction to recognise netting	• *external* : **for each service request,** ad-hoc reporting rules (for example between Netting and Structure&Size Limits) • *internal* : **for each core capability,** (ActiveX link to) ad-hoc reports (between the core capability (e.g. Structure&Size Limits) and its managing object, (e.g. Limit Management) [7]

[1] Summary report has a service name only; detailed report also give "Object Returned" and "Arguments" as specified in appropriate Firmament Framework form

[2] Word 'output' is meant to exclude management reporting within the process

[3] Target KPI=100, measurement of actual=90 as of today, exception rule=80 i.e. generate exception report if actual below 80

[4] Here we differentiate between the trigger (service request and any accompanying data) and the info ('inputs') required to provide service response

[5] Word 'input' is meant to exclude management reporting within the process

[6] See footnote 3

[7] For example, when enforceability of netting in Malaysia changes significantly not only does this impact the outputs from Structure&Size Limits but Limit Management may request a major review of Malaysian limits

SUBSTITUTE SHEET (RULE 26)

Table 3 (Continued)

Generic Report Type	Example Reports for Board	Example Reports for Group Finance	Example Reports for Limit Management	Directly or indirectly implied changes to the meta-model
SLA	• performance against KPI e.g. summary data on HR (such as turnover)	• performance against KPI	• performance against KPI	• **for each service request,** includes KPI, charging mechanism, conflict resolution etc, actual costs and fees (maybe an ActiveX link to and/or fields in the form)
Service consumer feedback on provision of service			• feedback on credit problems / mistakes / complaints • limit utilisation	• **for each service request,** (ActiveX link to) feedback
Internal reports on provision of service		• established, utilised and undrawn funding lines [1]		• **for each service request,** (ActiveX link to) report
Internal process costs s and local budgets				• **for each process,** (ActiveX link to) budget *v* spend *v* earned value
Issues	adhoc items of major sensitivity or importance (eg regulator plans disciplinary action)			• **for each service request,** issues and their priorities
Improvement suggestions/projects [2]	• progress on major initiatives			• **for a service or a service provider,** we list suggestions and any authorised projects (incl. budget, timescales).

[1] **An example of linking the process model in Firmament Framework tool with the core Firmament software**

[2] **An improvement project can be A.** change to an existing service, **B.** introduction of a new service within an existing service provider, **C.** introduction of a new service provider. Propose to use a colour scheme to distinguish between **a.** currently no projects, **b.** project is, say, one of red/amber/green.

SUBSTITUTE SHEET (RULE 26)

Data Linking

A further major benefit of the present invention is that data may be fed back from the actual operation of the business into the business model. Thus, documents may be associated with particular business process objects within the model.

An example of this is for a customer business process object to be linked to data indicating the number of customers using the business. This allows any member of the organization to access the model and obtain information regarding the number of customers the bank currently has. In addition to this because there will be a number of different customer business process objects within the entire bank, this allows the values to readily be broken down for individual business process objects in the bank or for the bank as a whole.

By regularly monitoring this data, this allows efficiency of the business to be determined. Thus for example, the expenditure of the company will be available from the model as well as indications of the profit. This allows the operation of the business to be monitored so that improvements to the business can be implemented.

This illustrates an important point that business models not only need to be summarized and reported (e.g. for duplications and gaps), but that they can be used to introduce a structure into management reporting.

In the present invention, business processes and service requests are supported by forms. These can be ActiveX documents which allow a "physical" link between a process model and the day-to-day operation of that process to be established. In this the invention becomes the control center of, say, a dynamic Bank which designs process improvements, measure the KPI's and, based on the measurements, proposes further improvements.

Further Uses

The present invention can also be advantageously used when considering business systems operating on communication network systems, such as the Internet. In this example, the business system could be considered as a network of service-to-service providers with each service provider providing a number of related services. In this case, each service provider could be defined as a respective business process object in the design of the business.

In addition to this, additional business process objects can be provided which represent consumers of the services. These may themselves provide additional services or simply be general public customers.

From the above described implementation of the invention, it will be realised that consideration of the statecharts of the various business process objects will allow the Internet based service providers to be easily implemented. Furthermore, the service implementations of the service providers can often be defined as Enterprise Java Beans which allows services to be readily provided.

This system will also allow private and public services to be defined with the non-public services only being available to a limited number of service providers and/or consumers, or alternatively the services may be internal within a service provider itself.

This type of malleable business design allows the designed business system to be highly flexible so that the system provides a highly reconfigurable set-up where services are out-sourced, co-sourced, in-sourced and/or shared as appropriate. This reflects the dynamics of the ever changing business relationships which typically occur for Internet based businesses. This system therefore allows alterations in the industry to be readily integrated in to the business model, whilst improvements in the business model can readily be propagated into the working business system.

Examples of Internet business-to-business services are the credit risk services that banks are likely to want to

make available to their clients. This will tend to be along the lines of the trades the client has, what their current value is, what the current collateral hold against them is, what netting arrangements are in place and what limits are placed on the trades. Accordingly, as will be appreciated by a person skilled in the art this allows clients to obtain information directly from the bank in accordance with the defined business model.

Specific Example

A specific example of the implementation of the invention to determine a method of operating a business will now be described.

In the following example, the intention is to build a capability to administer credit derivatives throughout their life cycle. It is necessary to generate a complete process definition along with associated user documentation, as well as to automate a large part of the process. In this example, it will be assumed that a business process object for administering effects and IR trades is already stored in a database, along with an associated state chart diagram which is shown in Figure 7.

Firstly, it is necessary to carry out a high level analysis of the service requirements needed to administer credit derivatives. This involves analyzing the requirements to administer credit derivatives and determining therefrom the actions that are required for the associated credit derivatives business process object to carry out. The requirements of the credit derivatives object are then compared to a database of alternative business process objects and it is revealed that there is a high level of similarity to the FX/IR trades business process object. The main difference between these business process objects is that additional features are required in the credit derivative business process object.

Accordingly, the credit derivative business process object (hereinafter referred to as the credit derivative

object) is assumed to be a specialization of the FX/IR trade business process object (hereinafter referred to as the FX/IR trade object), as shown in Figure 5A.

The use of a predetermined business process object will allow the clients to benefit from uniform handling of FX/IR and credit derivative trades as well as being able to employ the high level of automation that is currently present in the FX/IR trade object without having to generate the system from scratch.

However, in this example, the potential clients of the credit derivative products decide that they also want the rate fixing of the credit derivative to be automated. Accordingly, in addition to extending the FX/IR trade object to handle "credit events" as required by credit derivatives, it is also necessary to adapt the FX/IR trade object to allow the software automation of "rate fixing". This is shown in Figure 6. It will of course be appreciated that other requirements may be placed on the credit derivative business object.

The first stage is to access the statechart of the FX/IR trade object which is shown in Figure 7. This statechart sets out the different states of the business process object with event response actions indicating services provided by the individual states.

In order to proceed further, it is necessary to analyze the business events that are required to provide for the handling of credit derivatives. This includes determining not only whether any new states need to be added to the statechart shown in Figure 7, but also to determine whether it is necessary to modify any of the existing states.

In this case, in order to handle credit events, the following modifications are required:

New State:

- Assessing Impact of Credit Event (AICE)

New Event in state Trade Management:

- credit event (causes transition to AICE)

New Events in state AICE:

- further credit event (causes no state change)
- ignore event (causes transition back to Trade Management)
- cancel (causes transition to Trade Management)
- exercise (causes transition to Trade Management)
- handle event which permanently changes deal (causes transition to Trade Management)
- restructure in response to credit event (causes transition to Amending/Restructuring)
- expires worthless (causes transition to Ended)
- full cancellation with no payments due (causes transition to Ended).

New Events in state Amending/Restructuring:

- credit event (causes transition to AICE)

Analysis of these goals shows that whilst the initial plan was to derive credit derivatives from the FX/IR object there are certain implications in this structure which must be considered. The most notable of which is that any future changes to the FX/IR trade object would be directly inherited by the credit derivatives object. It can be readily foreseen that certain changes to the FX/IR trade object would not be desirable in the case of the credit derivatives object.

Accordingly, a new more sophisticated relationship between the two business process objects is proposed, as shown in the object hierarchy diagram shown in Figure 8. In this case, a more general trade object is generated, which at the current moment is simply a direct copy of the FX/IR trade object. The FX/IR trade object then becomes a specialization of the trade object without any changes. The credit derivatives object is similarly a specialization of the trade object.

Effectively, the services provided by the trade object are specialized to administer credit derivatives, with some changes and extensions where necessary. In the case of the FX/IR trade object no changes are needed. It should be noted

that initialization can redistribute people and systems responsibilities for provision of inherited services. For rate fixing, however, both manual and automatic services are required.

Returning to the statechart diagram shown in Figure 7 which now applies to the trade business process object, it is now necessary to make the modifications outlined above to ensure that the credit derivatives object can handle credit events as well as allow for the automated rate fixing.

The first modification is to add the additional state of accessing impact of a credit event (the AICE state) and then link this to the current states by a number of appropriate event response actions. This results in the statechart shown in Figure 9.

Next it is necessary to modify the existing states of the statechart to allow for the automated rate fixing capability. In order to implement the changes that are required to trade management state, it is necessary to derive a more detailed representation of the state to determine where the changes must be incorporated.

Accordingly, a second level statechart is determined for the credit derivatives trade management object (it will be appreciated that each state of the statechart is in itself an object which is an instance of the credit derivatives business process object). This second level statechart is shown as a diagram in Figure 10 and is derived either from an analysis of the capabilities of the trade management object, or preferably by considering the second level of the statechart for the trade management object of the more general trade object.

An analysis of the second level statechart sub-diagram shown in Figure 10 reveals that the rate fixing capability is a capability of the managing cash flows object. Accordingly, a third level statechart is produced of the managing cash flows object, as shown in Figure 11.

In this case it can be seen that the subsequent fixings due event response action will need to be performed in a

different way to if the system is to provide automatic rate fixing as opposed to using a manual system. This represents a polymorphism of subsequent fixings due event response action.

In view of the relationship shown in Figure 8, the rate fixing related services can be generalized to the general trade object if desired. Such an action will then allow the FX/IR trade object and the trade object to inherit the automation of rate fixing. This is a useful side effect of the present invention in that modifications made to existing business process objects are propagated to similar business process objects thereby increasing the efficiency of the entire business.

This improved approach to automated rate fixing is represented in a modified object hierarchy diagram as shown in Figure 12. As shown the automated rate fixing due is now a feature of the trade business process object and is therefore inherited by the FX/IR trade business process object.

The next stage in the process is to apply traceability between high level business design and the software implementation level. This allows software from the trade business process object to be used in the credit derivatives business process object.

In this example, the refinement of the fixing business sub-process, taken from the business diagram for trade administration is demonstrated. This refinement process also establishes a close relationship between core capabilities diagrams and state chart diagrams. Figure 13 shows a list outlining the relationship between the statechart diagrams of different levels.

Figure 14 is a high level core capabilities diagram for trade administration which shows core business activities which are defined in more detail in subsequent diagrams. The starting point here is Trade Administration which is an identified function of the Market Area introduced by earlier workshops. It is a front/middle/back office capability for

handling trade definition and manipulation.

Figure 14 shows that the main business activities (core capabilities) for Trade Administration are: Booking, Amending, Trade Management, Assessing Impact Of Credit Event, Valuation and Analysis. It further shows the interaction between these capabilities and other system areas (whether business or infrastructure) - for example the interaction with Confirmations.

At this level the service requests are indications of business intent rather than of software implementation. The subsequent refinement of the core capabilities listed above will start to show the exact adoption of these services by business objects. This is particularly true for interaction between Trade Administration as a whole and Counterparty, Confirmations, Financial Control incl. Accounting and others.

Each of the core capabilities of Trade Administration (those objects surrounded by the "box") can be seen as a state for Statechart diagram for Credit Derivatives Administration which is shown in Figure 5. For example, the Booking core capability maps to the Booking state on the Statechart, Trade Management core capability to the Trade Management state, and so on. The only exception is Valuation and Analysis, which exists as a (major) event on the statechart. In general, states "handle" a number of events, this particular event can occur in every state so the transitions to/from Valuation and Analysis would have cluttered the diagram without adding much value.

Unlike Service Implementation diagrams, for overview purposes and completeness it is acceptable to show requests between two objects neither of which are a core capabilities (e.g. provide Treasury/LIBOR market data from Fixing Maintenance to Market Data Management).

For the purposes of this document, Trade Management will be the core capability / state concentrated on from this point. The business responsibility of Trade Management is to manage the trade - whether this is looking at an individual trade cash flow or the trade as a whole. In this sense, the

Trade Management core capability also represents a state within the trade lifecycle: the trade has been defined and dealt through the Booking process; it is now being managed.

That, in turn, requires additional capabilities.

The core capabilities identified are: Cash flow Management (a grouping of capabilities that apply to individual cash flows), Manage Full Cancellation, Option Exercise, and Manage Payments Due. These core capabilities can also be modeled as event managers on the Trade Management sub-states of the second level statechart shown in Figure 10.

Figure 15 is a diagram of the second level core capabilities of the trade management core capabilities of Figure 14. This shows more precisely how service requests to and from the Trade Management object (e.g. interaction with the "external" capability to manage Counterparty) have now been delegated to these core capabilities. For example, Payments and Payments Netting Management provides a service to Manage Payments Due.

The next step, as set out in Figure 13 is to examine the cash flow management core capability in more detail. In this case the sub-processes involved are manage cash flow payments due, manage manual first fixing due and manage subsequent fixings. A diagram of the core capabilities of cash flow management is shown in Figure 16 which represents the third level core capabilities. Further to this diagram, it is now necessary to examine the manage subsequent fixings capability further and accordingly, the fourth level core capabilities are determined as shown in Figure 17. This is a core capabilities diagram of the manage subsequent fixings capability.

It will be appreciated that in the case in which the trade business process object has been defined in sufficient detail, these core capabilities diagrams can be obtained directly from the core capabilities diagrams of the business process objects. However, if no such core capabilities diagrams have been derived, then the diagrams must be

generated from scratch by a consideration of the services provided by each of the capabilities and the way in which the capabilities interact.

The sub-processes involved in Manage Subsequent Fixings are: Calculate Fixing Price/Rate, Calculate Settlement Amount, Apply Rate/Price Fixing. In all of these core capabilities the actual implementation is not relevant as yet, although it will be shown at a lower level primarily through Service Implementation diagrams or occasionally through further Core Capabilities diagrams.

Each of the sub-processes has been represented as an object, although in the "final" implementation these may become services on a cash flow. On the one hand it is important to capture and reflect the business process for doing something rather than trying to make implementation decisions too soon. On the other hand, the core capabilities (groups of services) are typical units of reuse rather than individual services. Of course, it is perfectly legitimate to consider rate fix to be a service of Cash flow Management.

Shown in Figure 17, the rate fix is a service provided by the apply rate/price fixing object. Accordingly, the apply rate/price fixing object is the controlling object which manipulates and drives the calculate fixing price/rate and the calculate settlement amount objects.

The implies certain mutual responsibilities between Calculate Fixing Price/Rate, Calculate Settlement Amount and Apply Fixing. The service rate fix, corresponds to the business event subsequent fixing due shown in Figure 10.

Figure 18 shows the Service Implementation for the service rate fix indicating the typical sequence of events and interaction with other objects required to complete the service. This demonstrates how Apply Rate/Price Fixing delegates its responsibilities to Calculate Fixing Price/Rate, Calculate Settlement Amount, Amount, Rate and Confirmations to provide the service rate fix. The service is meant to be implemented in Java and triggered by subsequent fixing due event, see the Statechart diagram.

Indirect dependencies on other objects should be shown on separate service implementation diagrams. The closer we get to software implementation, the more important it is to understand that a service request to an object (for example update to Amount) must be met by services of the same name. In this case object Amount should have a service called update.

Business events defined above will have their contracts defined in more detail on the associated Firmament Framework form, an example of which is shown in Figure 19.

Guard:

Initially the guard was "is fixing due", however, the current underlying assumption is that this process will be automated through the use of a Fixing Timer. This has lead to the redundancy of this guard. The current guard is a Boolean expression to check if the trade has already been fixed (e.g. manually, so we do not want the rate to be overwritten).

Event:

The transition event in the Statechart diagram is subsequent fixing due, (see Figure 11: Statechart diagram for Managing Cashflows), where the Event Type is Time Event. At this stage, the interfaces to (Fixing) Timer are undefined so there are no services available. Future design on the Timer should allow completion of the form, specifying the service generating event.

Action:

The Action invoked is the rate fix service - which maps to the business service supplied by the object Apply Rate/Price Fixing.

The key requirement is that states, events and response actions on the Statechart diagram must be provided by business objects within the design.

Appendix A

Object Design

An object design is a blueprint for the provision and rapid replication of a group of related business services. Good designs enable efficiency, re-use and consistency. Initially they result from a systematic analysis of business 'requirements'. In a mature effective business, new requirements do not always necessitate fundamentally new designs, as many of the existing designs can be re-used with or without improvements, thus allowing a collection of predetermined business process objects to be developed and re-used during subsequent business analysis.

Business Process Object

A business process object is an enactment (instance) of a business object design. Typically, an object design will have a number of enactments, each of which is capable of conducting responsibilities specified in the design.

A well designed assembly of communicating business process objects is a business process object in its own right, usually called 'business (sub)process'. For example, Savings Account management.

A business process object is represented schematically in Figure A. In this example the business process object is a particular branch of a bank which is implemented using a branch object design.

Encapsulation

The term refers to the fact that the business services of an object design should be complete and void of duplication, and be void of features which are better suited to other object designs.

The data and implementation details of a business process object, although potentially visible, are not intended to be directly accessed by other objects. This is to allow for the propagation of changes, to promote re-use, reduce the maintenance cost and minimize the risk of

improvements. There are varying levels of this 'information hiding' which allows for selective access to details for those 'who need to know' (so called public, private and protected business services, as set out in more detail below).

An example of 'information hiding' is in the case of a staff working roster which is essential for the successful operation of the Branch. Typically, however it is not advertised to the customers and would therefore be hidden from a customer object. Similarly, the deployment of Branch security services is not meant to be widely understood.

An example of a situation in which features of a business are better suited to alternative business process objects is for the business process object for operation of the Mail-room in a Bank. The mail-room manages internal and external mail and nothing else. As a result, features such as the catering facilities should be managed by another business process object.

An example of completeness and 'information hiding' is in the case of a payroll which generates pay slips and initiates payments. In this case most of the necessary detailed implementation (e.g. calculation of National Insurance/Tax deductions, Pension contributions, returns to the Inland Revenue) are not normally published to the Bank staff who only receive summary information.

Business Services

A business service is a clearly defined, non-redundant, sharply focused ability of a business process object to provide capabilities. If a capability is provided to other objects the capability is known as a "public" service whereas if the capability is used to assist the provision of other capabilities of the same object it is known as a "protected" or "private" service.

A business service may be either centralized (static) or distributed as well as primary or acquired (as defined in more detail below).

Service Request / Delegation

Public (but not private or protected) business services of a business process object can be requested by other objects. This is done by issuing a service request named after the existing service. An example of this is shown in Figure B in which a request for approval of a loan is made to a loan application service.

Centralized Service

A single business service which relates to all enactments of an object design is known as a centralized service. Such centralized services are rare but do occur. One example, would be the production of a weekly report calculating the Total number of successful customer engagements across a network of Bank Branches.

A Centralized Attribute is a special case of a Centralized Service, for example the Total number of Branches in the Branch network.

Distributed Service

A business service which relates to a single business process object is known as a business service. Typically, it operates on the data encapsulated in the owning object and, perhaps, on some centralized (static) data, but not directly on the data of another object.

For example, the responsibility to obtain customer signature on a Loan Application, or to verify customer ID is distributed to all branches. Nevertheless, the customer data is managed centrally.

A Distributed Attribute is a special case of Distributed Service. This is information which is managed by and is responsibility of an individual business process object, for example a Bank Branch Diary.

Primary Service / Core Capability

A core capability is an encapsulated business service. Thus it is an encapsulated auxiliary business process object

('attribute') together with all its associated business capabilities. For example, for a Branch service to accept cash and credit a customer's account also utilizes services of an Account object.

Primary services of an object design can be brand new features, or passed down ('inherited') from other object designs. Alternatively primary services can be features which are specialized versions of passed-down business services, see below.

Acquired Service / Non-core Capability

Some business process objects out-source one or many of their responsibilities to the named external (i.e. not encapsulated) objects and are therefore known as non-core capabilities.

An example of a non-core capability is shown in Figure C which shows stationary being obtained for a branch account object from an external supplier.

Private Service

An auxiliary or transient service which is visible to other business process objects but is not intended to be used by them is a private service. For example, Security arrangements are vital for operation of each Branch but, typically, branches do not offer them as a service to other business units of the Bank.

Protected Service

A business service intended to be used only by the owning business process object or specializations thereof is a protected service. For example, Security arrangements for A.T.M. cash can be specialized to cover cash deposits but probably not to cover Catering cash.

In absence of object designs specializations, protected services are no more accessible than private services.

Public Service

A stable and permanent business service designed to be used by other business process objects is a public service. Typically, public services are implemented through a number of fine grain private or protected services. For example, Telephone Banking currently offers a relatively narrow range of services to the customers but requires a much wider people and systems infrastructure.

Design Specialization / Inheritance

A technique for creating more detailed object designs from the existing, more general designs ('a base class') is known as specialization. This technique promotes re-use, consistency and integrity of object designs. For example, the service to provide financial advice for a new product is a specialization of the general capability to provide financial guidance to the customers.

Figure D shows the situation in which the public and protected services of an account object are inherited by a more specialized mortgage account service. This inheritance can be with or without changes.

Specialization is the foremost business design technique to manage the ever-changing market conditions. It should be also seen as extending or adding value to the general object designs without affecting the common parts. Note that specialization can re-distribute people and systems responsibilities for provision of passed-down services.

In the present invention Object Hierarchy diagrams are used to represent design specializations.

Design Generalization / Inheritance

Generalization is a technique for creating more general object designs from the existing, more detailed designs. This technique promotes re-use, consistency and integrity. For example, the service to provide total financial guidance to the customers is a generalization of the more narrow capability to provide financial advice for a new product.

Another view is that this is abstraction, i.e. identification of core capabilities.

In the example of Figure D, some services of the mortgage account object can be generalized and used by the more general account object.

Object Hierarchy diagrams, which are explained in more detail below, are also used for design generalizations.

Specialization of Internal Capabilities

The specialization of internal capabilities enables internal re-use, with or without improvements, of services which were passed down from a more general design. The point here is that services inherited this way are not being made directly available to other business process objects (even though they may have been accessible from the enactments of the more general design). For example, a policy concerning security over cash is passed down from a general policy object but would have to be specialized if it were to be used by an object dealing with a particular type of risk (e.g. A.T.M. cash).

Specialization of Public Services

The specialization of public services enables the provision of additional public services by re-using, with or without improvements, the public services passed down from a more general design. For example, 'Mortgage Lending may be specialized to support first-time-buyer mortgage lending.

Polymorphism

Different business process objects may respond differently to the same service request, yet the design of the object guarantees some level of consistency. This is known as polymorphism. For example, greeting customers can be consistent, yet customers' specific circumstances (e.g. a recent death in the family) should be taken into account.

Polymorphic objects may appear identical even when they do not have a single line of code in common. Thus giving both

the business clients and local service providers a considerable advantage: the former need not worry about the detail, the latter have the clarity of objectives without too many constraints.

For solutions vendors, the key to 'customer intimacy' whilst preserving consistency.

Very importantly, business process objects in the same object hierarchy have polymorphically consistent statecharts.

Inputs of a Service

The totality of information required for the provision of a business service is known as the inputs of a service. For example, completed application form is required for the service to open a new account.

Service Pre-condition

Service pre-conditions must be satisfied before a business service can be provided. For example, signing (by the customer) of an Application form. The act of signing changes the state of the Application object.

Service Deliverable

A deliverable service is output resulting from provision of a service. For example, a new A.T.M. card is generated when a new account is opened.

Service Outcome

In addition to providing the deliverables, provision of a service may change the range of services offered to the customer. This is known as the service outcome. For example, Account usage is restricted when the customer goes "missing".

Expressiveness

Expressiveness refers to keeping options open, for example, technical details of a new financial product.

Business people can drag&drop business objects and create products (without programmers involvement).

Appendix B

Diagram Types Overview

The following section provides details of the types of diagrams used in the methods of the present invention.

Core Capabilities Diagrams

Core capabilities diagrams are intended to give confidence that all external responsibilities of the business (sub)process undergoing improvements are being met. Core capabilities (see section Design Generalization / Inheritance) encapsulate a group of related services, for example, all event response actions which may be required in a process state. They also show mutual responsibilities between core capabilities, these are expressed as service requests (arrows pointing to service providers). At this stage of design we also identify major business events.

Statechart Diagrams

Statechart diagrams partition the business event space into manageable sub-spaces, where event response actions are services provided by business objects. Statecharts give engineering rigeur to business designs and sufficient precision to object hierarchies by enforcing behaviourial aspects of objects. They may also provide the first insight into Graphical User Interfaces as described further below.

Service Implementation Diagrams

Service Implementation diagrams give engineering rigeur to the design of individual business object services where necessary. They promote delegation, encapsulation and information hiding (an example of such a diagram is shown in Figure 4A).

Business Improvement Environment Diagrams

Business Improvement Environment diagrams define the scope of process improvements. They also identify external

and internal clients of the process, and the mutual responsibilities. It is important to note that the system users participate in the process undergoing improvements, whereas client processes and their users are "outside" (as explained in section Business Design).

Object Hierarchy diagrams

Object Hierarchy diagrams are used to highlight the requirements of inheritance and polymorphism and effectively represent the relationship between different levels of a business object designs. Inheritance is treating new situations as extensions of known situations. Polymorphism means agreeing what not how. It is important to note that polymorphism is an analysis / design issue and not just a programming language feature. It can reduce dependencies by an order of magnitude.

Ptech Distributed Class Models

Ptech Distributed Class Models capture design decisions regarding the distribution of object services within the target computing environment, e.g. between clients and the server.

The focus is on Core Capabilities diagrams, Statechart diagrams and Service Implementation diagrams.

The above diagram types form three complementary views of the business: Object Interactions, Object Hierarchy and Behaviour, and Distributed Object Design and Implementation, as described in the Section entitled Firmament Framework and as shown in Figure 1.

Each of the three views is important and no single one is sufficient. They are meant to be applied iteratively until the design is complete and consistent. Thus the present invention removes the usual ambiguity between top-down and bottom-up approaches to business design and change management.

CLAIMS

1. A method of designing or optimizing a business, the method comprising defining the business as a number of business process objects and their interactions using object orientated design techniques, each business process object being designed to provide one or more services in response to an event.

2. A method according to claim 1, defining the business as a number of interacting statecharts, each statechart representing a number of states of business processes or sub-processes.

3. A method according to claim 1 or claim 2, wherein the method comprises defining the business as a number of statecharts, each statechart representing a number of states within the business, the states being interconnected by events and corresponding event response actions, each state corresponding to a respective business process object and each event response action corresponding to a service provided by the respective business process object in response to an event.

4. A method according to claim 3, wherein the method further comprises determining the service implementation of at least some of the event response actions, the service implementation defining the procedures required for implementing the respective event response action.

5. A method according to claim 4, wherein the method further comprises defining at least some of the business process objects as respective statecharts so as to generate a hierarchical structure of statecharts.

6. A method according to claim 5, wherein the method of determining a statechart of a business process object comprises the steps of:

a. comparing the business process object to a number of predetermined business process objects;

b. determining if one of the predetermined business process objects can be related to the business

process object by an object-oriented process; and, either:

i. using a predetermined statechart of a predetermined business process object that is related to the business process object; or,

ii. generating a new statechart.

7. A method according to claim 6, wherein the predetermined business process object is a generalisation or a specialisation of the business process object.

8. A method according to claim 6, wherein the method of generating a new statechart comprises:

a. determining events to which the business process object must respond;

b. dividing the events into sets of related events, each set of related events being associated with a respective state of the business;

c. for each event, defining one or more event response actions for providing the required services.

9. A method according to claim 8, wherein the method further comprises incorporating the generated statechart and the associated business process objects into the model of the business.

10. A method according to claim 6 or claim 7, wherein the method of using the predetermined statechart comprises:

a. comparing the services provided by the predetermined business process object with the services to be provided by the business process object; and,

b. if necessary, modifying the selected statechart by performing at least one of the following steps:

i. adding in one or more new states, and interconnecting the new states to existing states using appropriate event response actions so as to provide the required services; or,

ii. modifying the event response actions of

existing states to represent changes in the services to be provided by the business process object.

11. A method according to claim 10, wherein the step of modifying the event response actions of an existing state comprise modifying the service implementation of the event response action such that the state provides the required services.

12. A method according to claim 11, wherein if the service implementation of the event response action cannot be modified, the method further comprises:

(1) considering the business process object of the state to be modified;

(2) determining a second level statechart of the business process object of the state to be modified;

(3) determining the states of the second level statechart diagram which are to be modified;

(4) for any states for which the service implementation cannot be altered to provide the required services, repeating steps (1), (2) and (3) to generate the nth level statechart diagram for which the service implementation of the state can be altered to provide the required services; and,

(5) modifying the service implementation of the event response actions of the states of the nth level statechart such that the modified statechart represents the services provided by the respective business process object.

13. A method according to any of claims 10 to 12, wherein the method further comprises:

a. considering the modified statechart and the predetermined statechart;

b. considering the effect of inheritance on the operation of the business process object as defined by the modified statechart and the predetermined business process object; and,

c. adapting the relationship between the business

process object and the predetermined business process object as required.

14. A method according to any of claims 4 to 13, wherein the service implementation defines one or more algorithms configured to perform at least a portion of the respective event response action.

15. A method according to any of claims 4 to 14, wherein the service implementation defines one or more sets of instructions indicating the tasks to be performed in order to perform at least a portion of the respective event response action.

16. A method according to any of claims 4 to 15, wherein the method further comprises determining the service implementation diagrams for a given event response action of a state, the method comprising:

a. comparing the business process object associated with the state to a predetermined business process object;

b. determining a service implementation for an event response action of the state of the predetermined business process object; and,

c. modifying the service implementation to allow the required event response action to be implemented.

17. A method according to any of the preceding claims, wherein the method further comprises determining the core capabilities of a state of the business, the core capabilities representing the business process objects and their respective interactions required for the state to be implemented.

18. A method according to claim 17, wherein the method further comprises:

a. comparing the core capabilities with a number of predetermined core capabilities for different business states corresponding to the predetermined business process objects; and,

b. modifying the core capabilities and the statecharts in accordance with the predetermined

core capabilities.

19. A method according to claim 18, when dependent on claim 16, wherein the service implementation is selected from the service implementation of one of the predetermined core capabilities.

20. A method according to any of the preceding claims, wherein the method further comprises generating statechart diagrams representing the states, the event response actions and the event for a given statechart.

21. A method according to any of the preceding claims, wherein the method further comprises generating core capabilities diagrams showing the business process objects and their respective interactions which represent the core capabilities of any state of the business.

22. A method according to any of the preceding claims, the object orientated processes including the use of inheritance.

23. A method according to any of the preceding claims, the object orientated processes including the use of polymorphism.

24. A method of presenting business data relating to a business according to at least claim 20 and claim 21, wherein the method further comprises associating business data with the statecharts and/or the core capabilities, thereby allowing the data to be viewed on the appropriate diagram.

25. A method of risk assessment within a business, the method comprising the steps of:

a. defining the business as a number of business process objects and their associated interactions using object orientated design techniques, each business process object being designed to provide one or more services in response to an event; and,

b. determining the risk of at least some of the business process objects not providing a service in response to an event.

26. A method according to claim 25, wherein the business is defined in accordance with any of claims 3 to 23, the method of determining the risk further comprising:

a. assigning a risk value to each event response action of each state of the business process object, the risk value being representative of the risk of the event response action not being performed;

b. defining a relationship between the event response actions of the states; and,

c. determining a risk value representing the risk of a business process object not providing a service, the risk value being determined on the basis of the risk values associated with each event response action and the relationship there between.

27. A method according to claim 26, wherein the method of assigning a risk value to a given event response action comprises determining the likelihood that the respective service implementation cannot be completed.

28. A method according to claim 26 or claim 27, wherein the method further comprises assessing the risk of a failure in a business by:

a. determining the risk of each business process object not providing a service in response to an event, for all events and all business process objects;

b. determining a relationship between the business process objects; and,

c. determining a risk based on the risk associated with each business process object and the relationship there between.

29. A method of designing or optimizing an Internet based business system, the business system being formed from a number of service providers, the method comprising defining the business system as a number of business process objects and their interactions using object orientated design techniques, each business process object representing a respective service provider and being designed to provide one or more services in response to an event.

30. A method according to claim 29, wherein the method further comprises defining a number of business process objects representing service consumers which interact with the business process objects representing the service providers to receive desired services.

31. A method according to claim 29 or claim 30, wherein the method is implemented in accordance with the method of any of claims 2 to 28.

32. A method according to claim 31, when dependent on at least claim 3, wherein at least some of the service implementations are implemented using Enterprise Java Beans.

Fig.A.

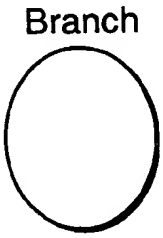


Fig.B.

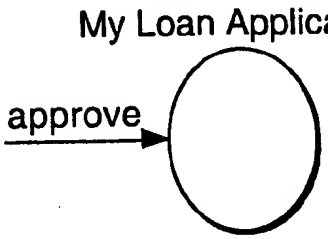


Fig.C.

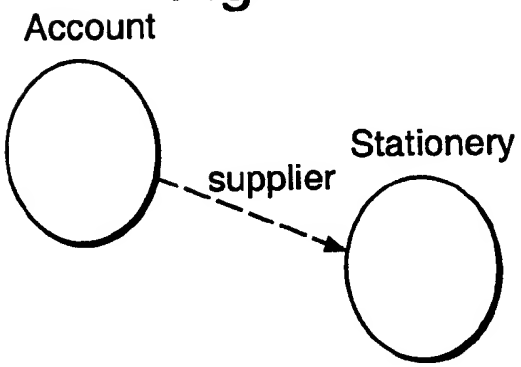


Fig.D.

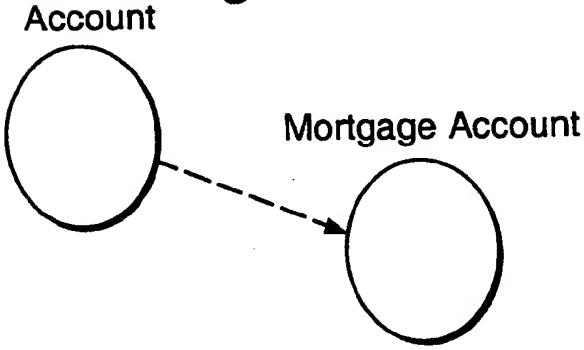


Fig.1.

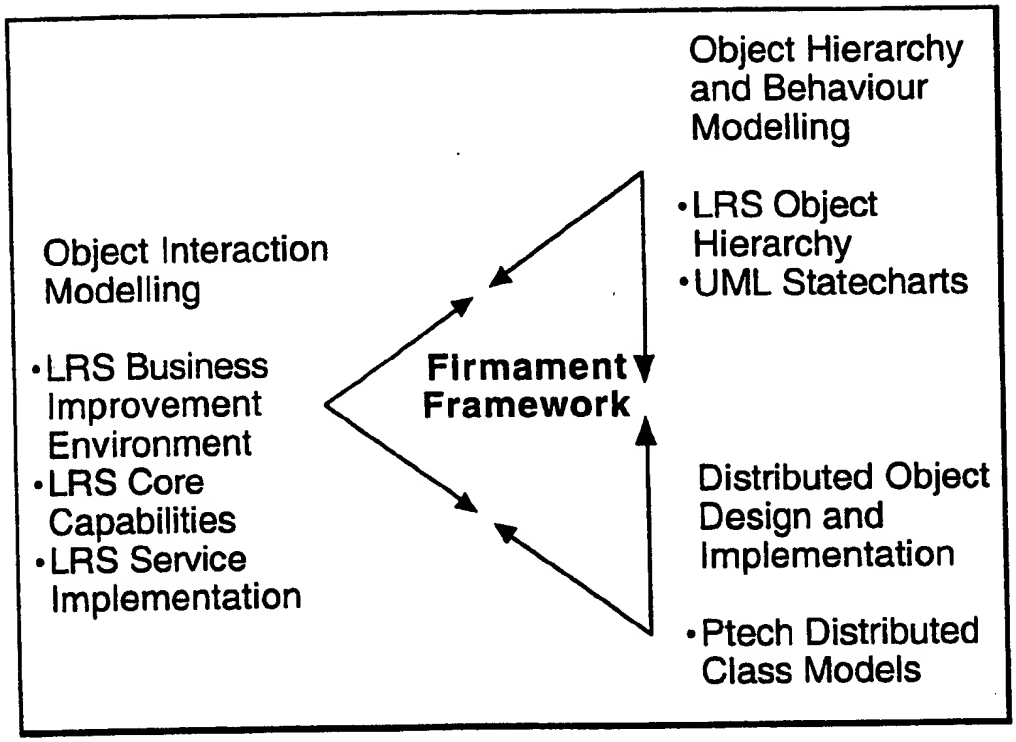

Object Hierarchy and Behaviour Modelling
•LRS Object Hierarchy
•UML Statecharts
Object Interaction Modelling
•LRS Business Improvement Environment
•LRS Core Capabilities
•LRS Service Implementation
Firmament Framework
Distributed Object Design and Implementation
•Ptech Distributed Class Models

Fig.2A.

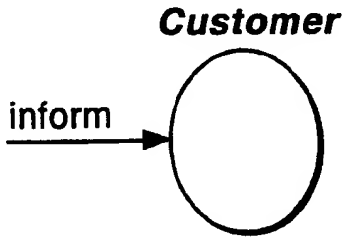

Customer
inform

Fig.2B.

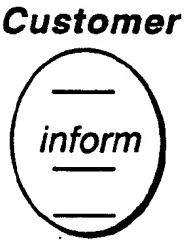


Fig.2C.

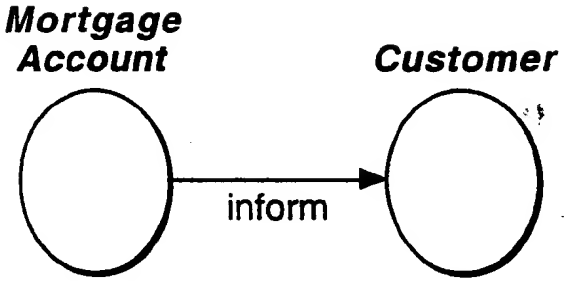


Fig.2D.

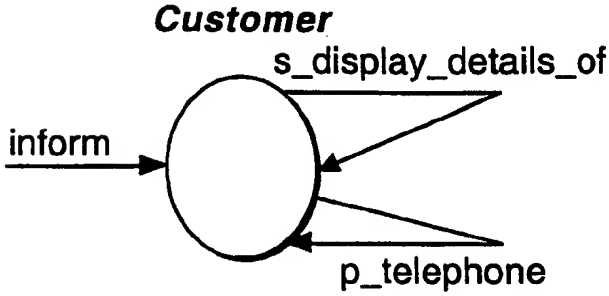


Fig.2E.

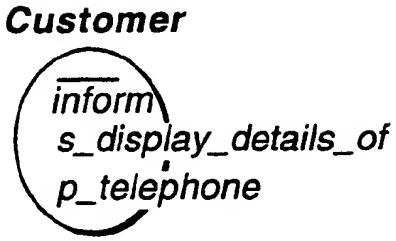


Fig.2F.

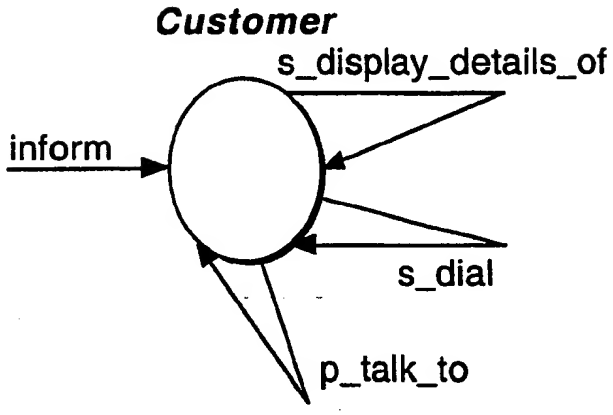


Fig.2G.

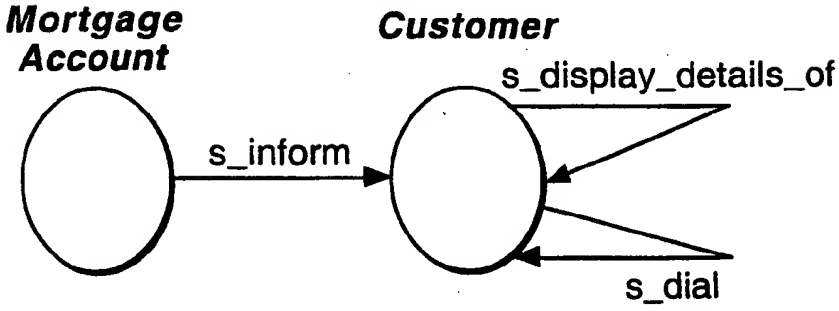


Fig.3A.

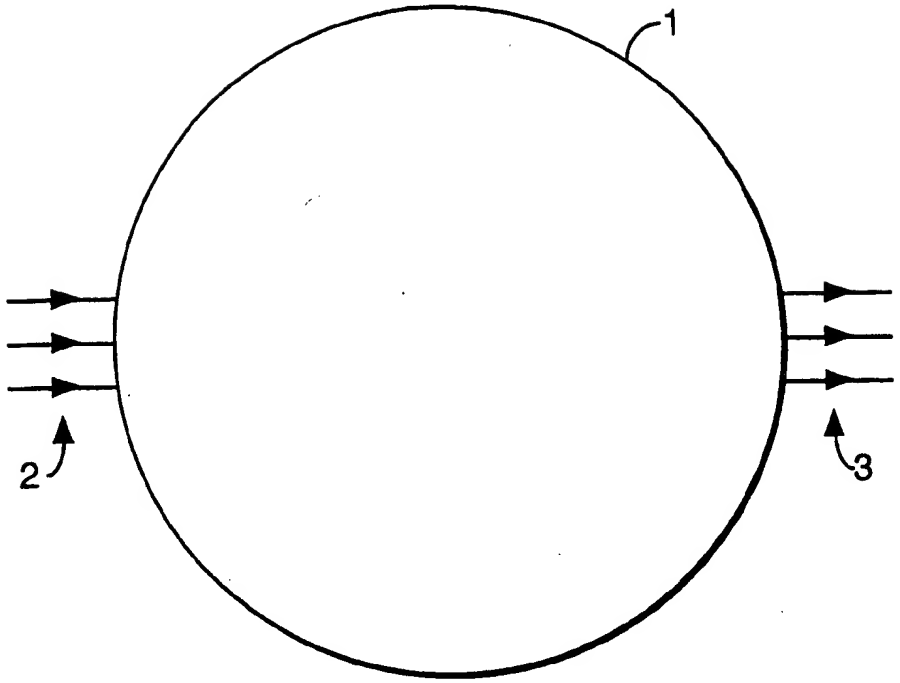


Fig.3B.

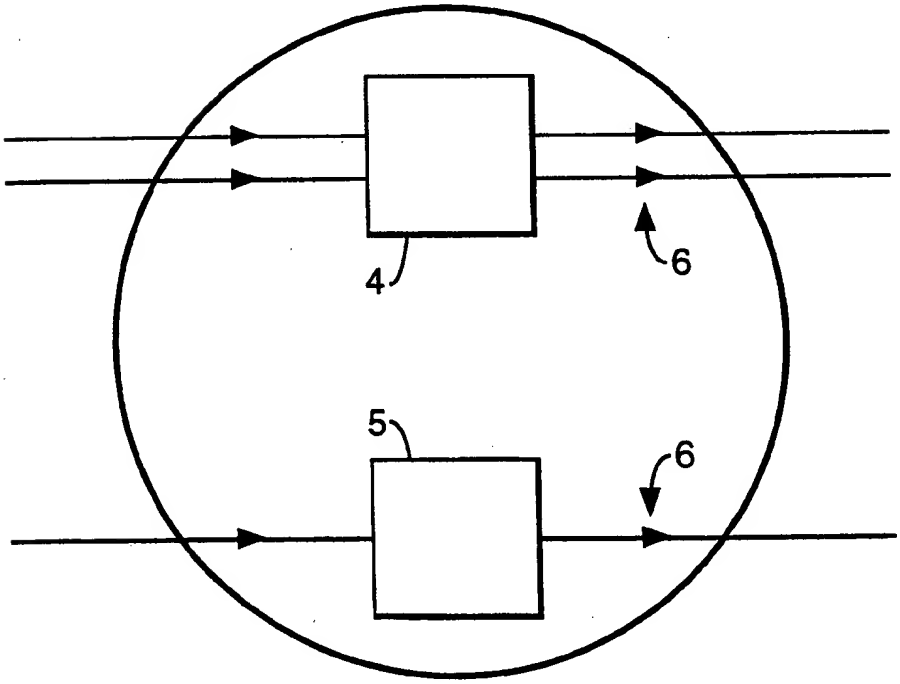


Fig.3C.

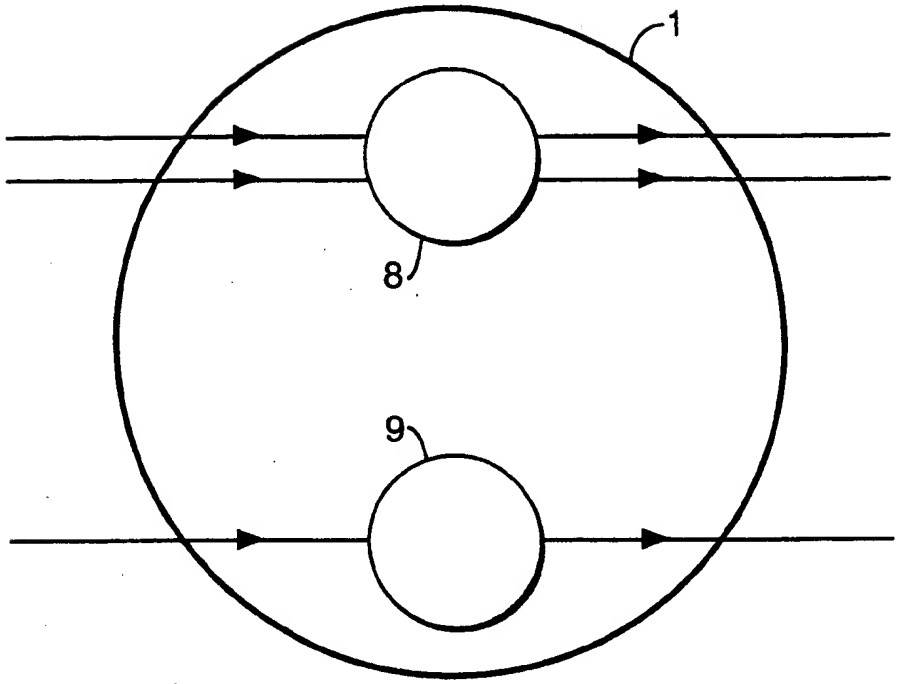


Fig.3D.

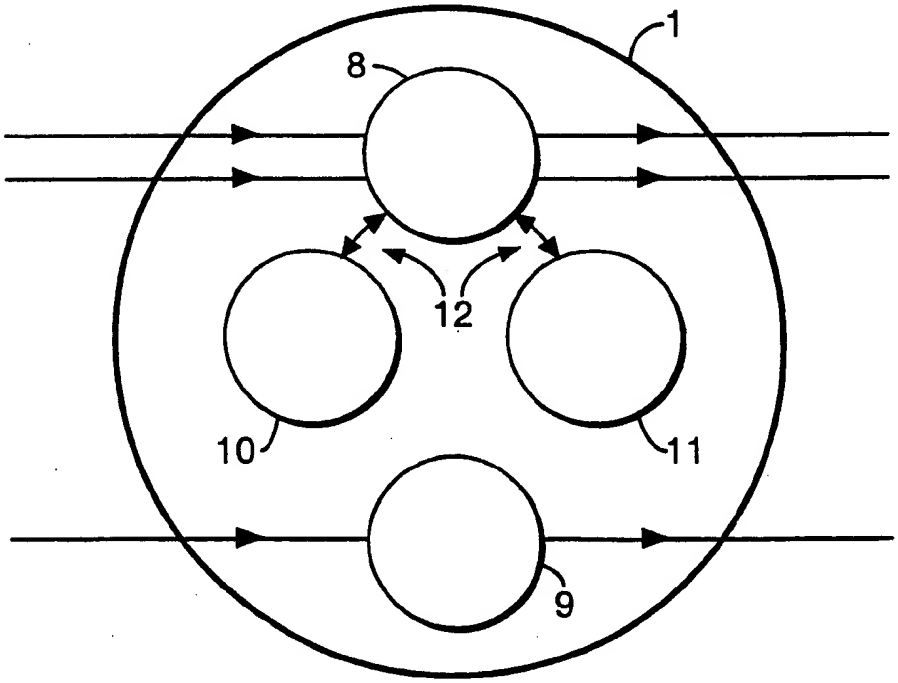


Fig.4A.

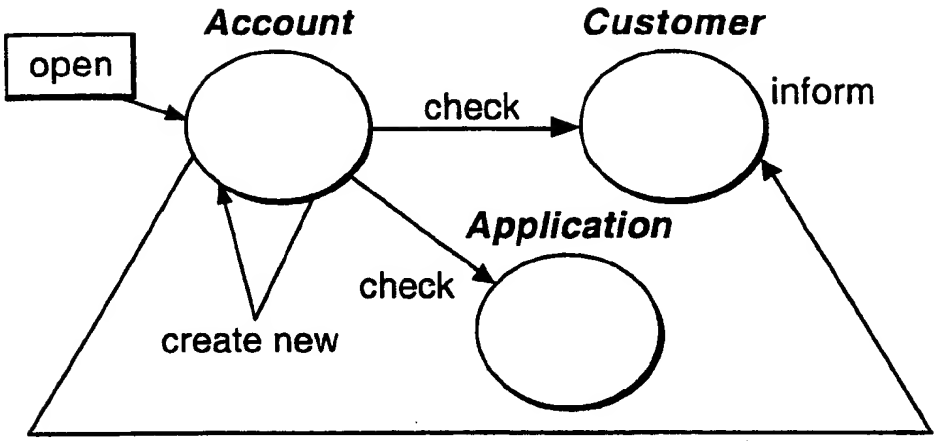


Fig.4B.

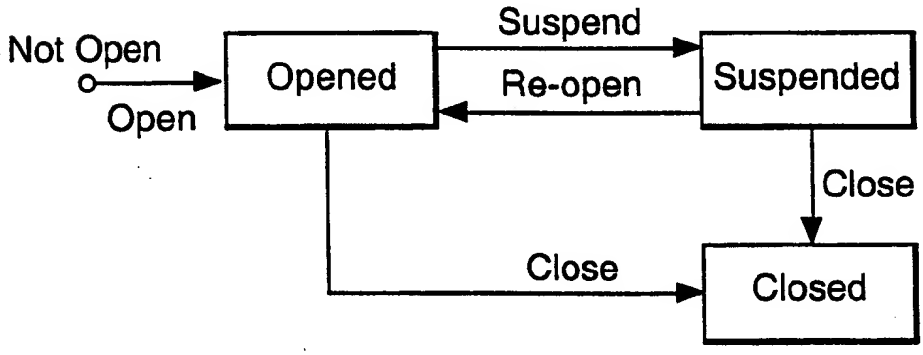
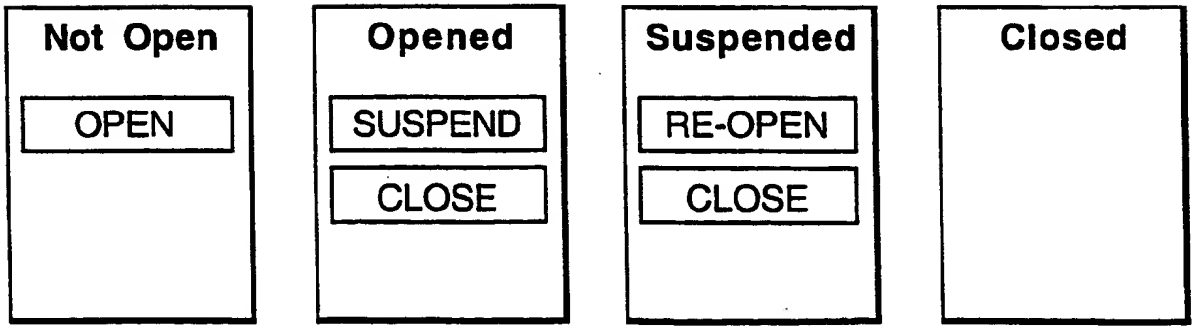


Fig.4C.



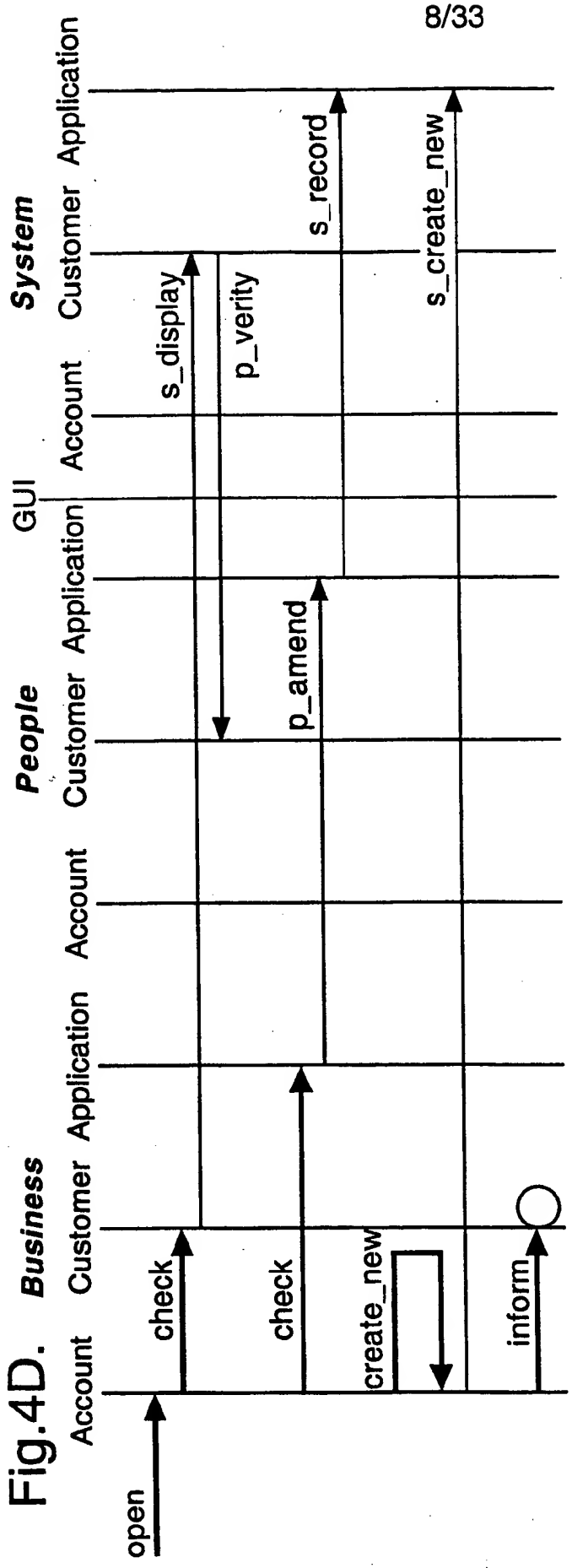

Business
Account
Customer
Application
Account
People
Customer
Application
GUI
Account
System
Customer
Application
open
check
check
create_new
inform
s_display
p_verity
p_amend
s_record
s_create_new

Fig.4D.

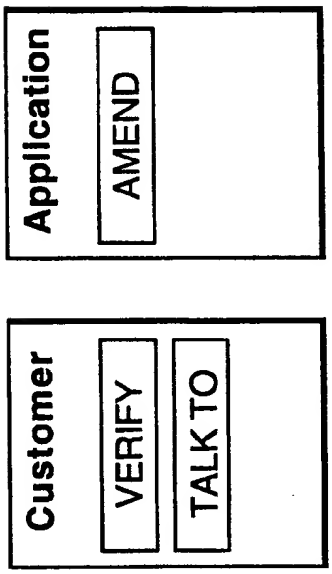

Customer
VERIFY
TALK TO
Application
AMEND

Fig.4E.

Fig.4F.

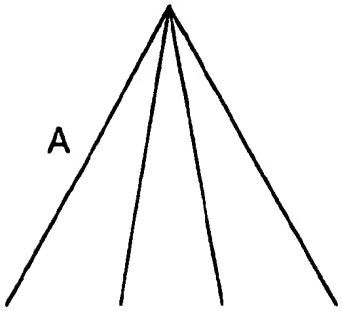


Fig.4G.

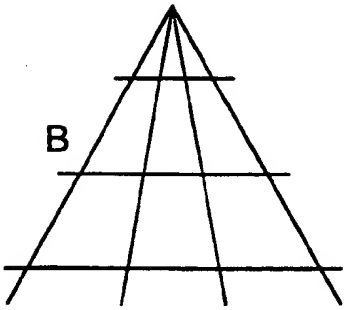


Fig.4H.

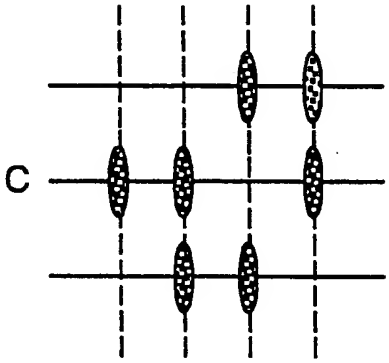


Fig.5.

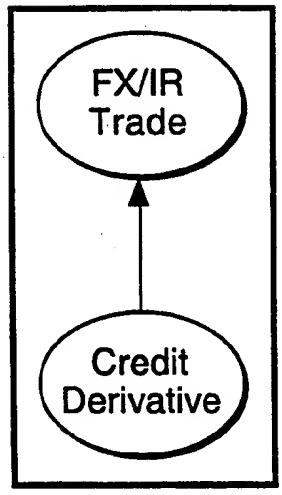
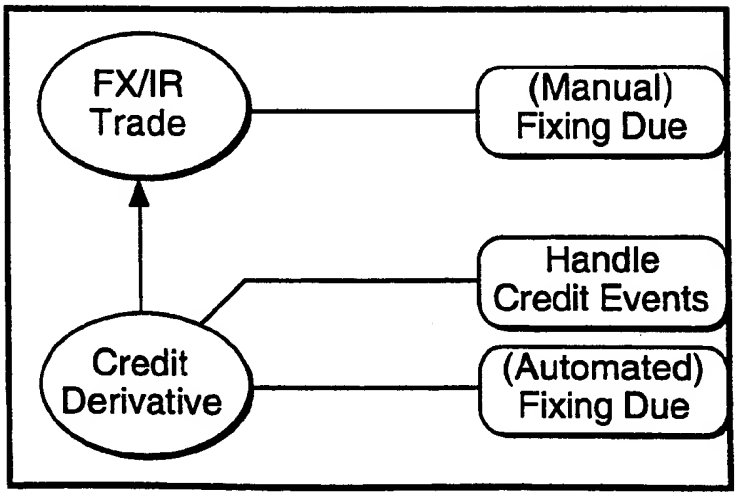


Fig.6.



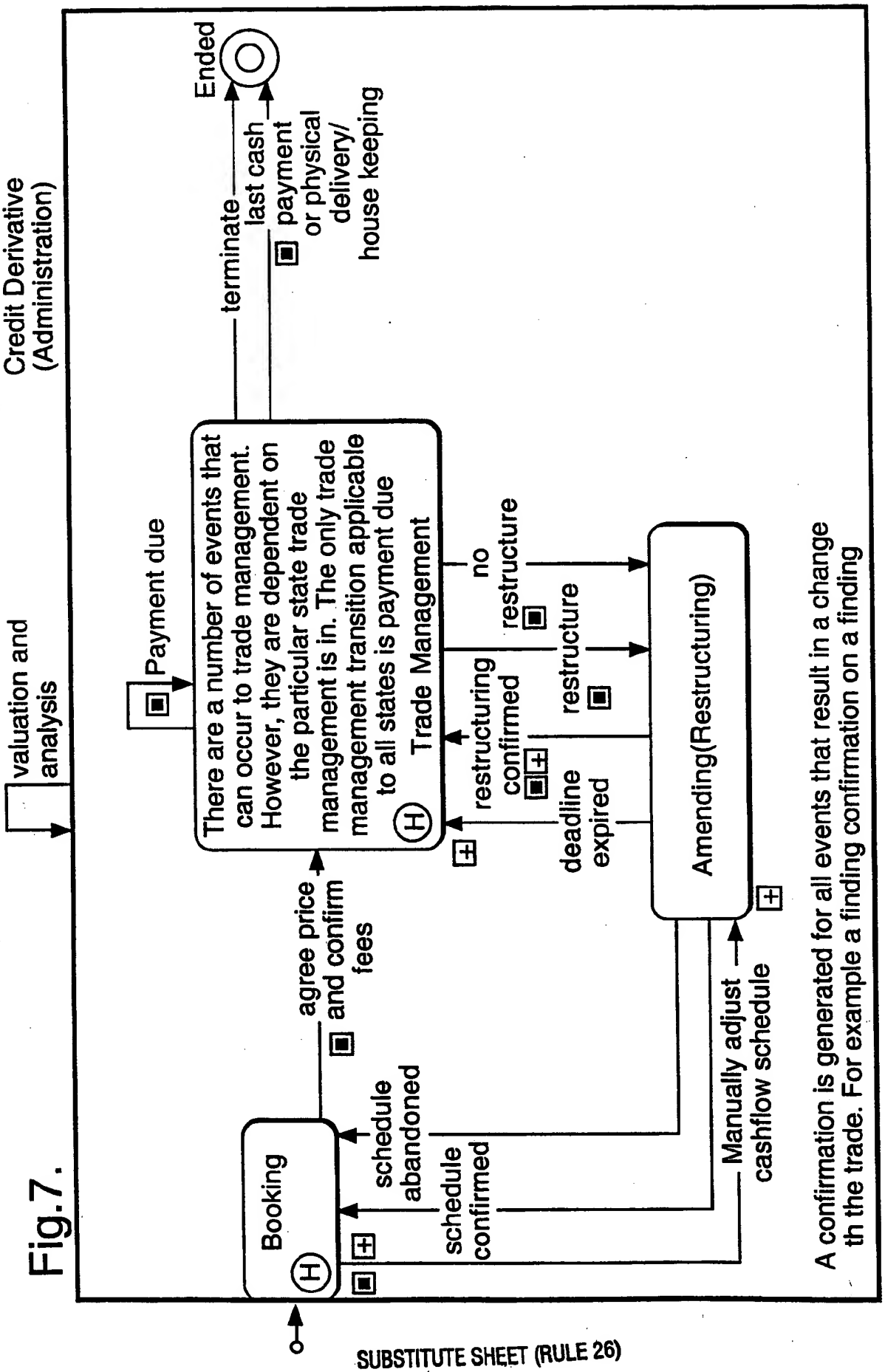

Credit Derivative (Administration)
valuation and analysis
Payment due
There are a number of events that can occur to trade management. However, they are dependent on the particular state trade management is in. The only trade management transition applicable to all states is payment due
Trade Management
terminate
last cash payment or physical delivery/ house keeping
Ended
no restructure
restructure
restructuring confirmed
deadline expired
Amending(Restructuring)
agree price and confirm fees
schedule abandoned
schedule confirmed
Manually adjust cashflow schedule
Booking
A confirmation is generated for all events that result in a change th the trade. For example a finding confirmation on a finding

Fig.7.

Fig.12.

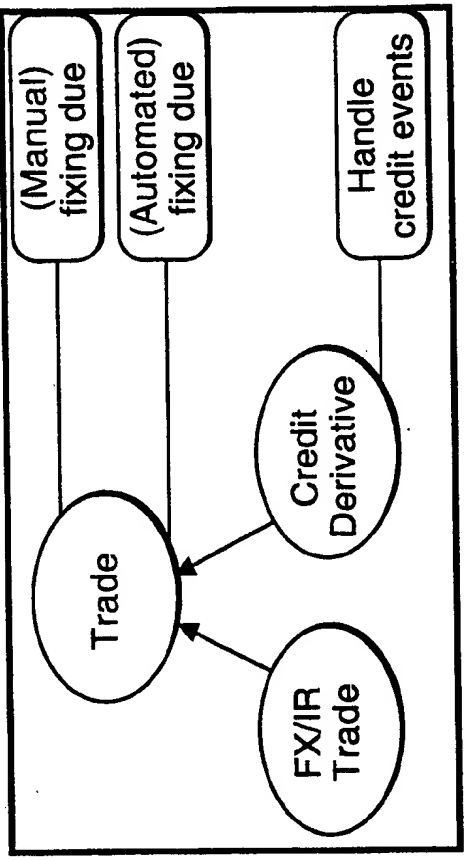


Fig.8.

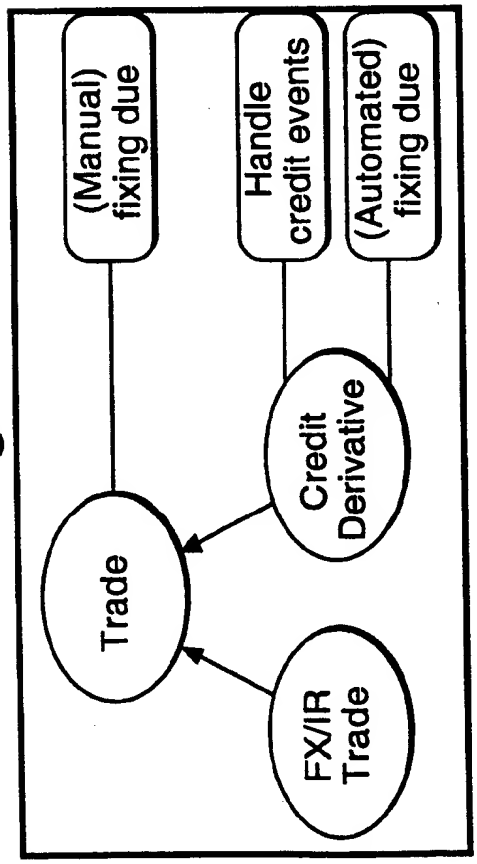
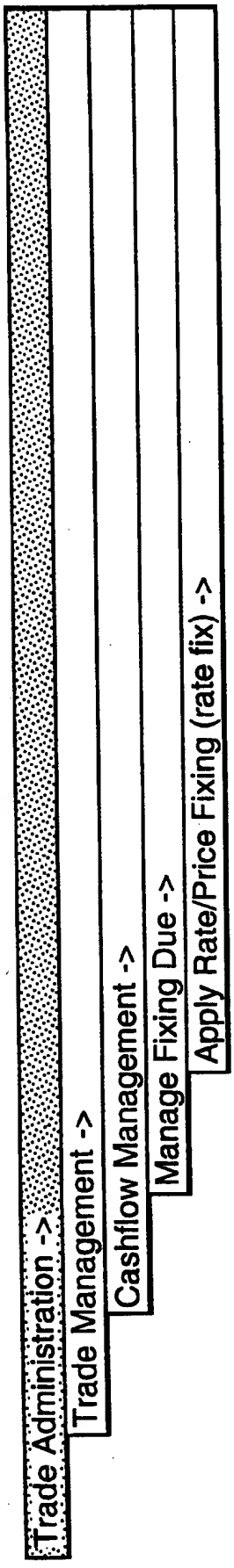


Fig.13.



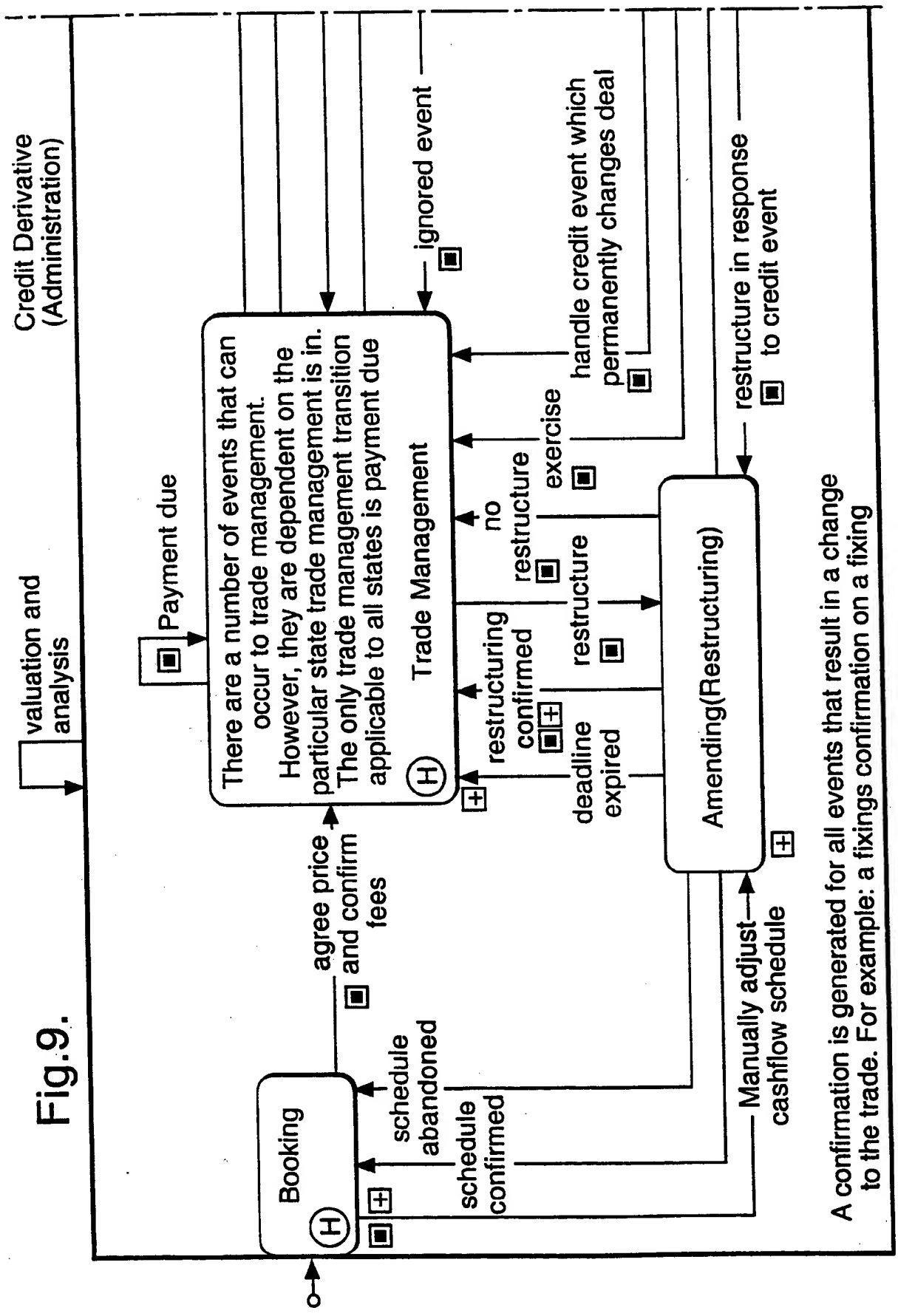

Credit Derivative (Administration)
valuation and analysis
Payment due
There are a number of events that can occur to trade management. However, they are dependent on the particular state trade management is in. The only trade management transition applicable to all states is payment due
Trade Management
ignored event
handle credit event which permanently changes deal
exercise
no restructure
restructuring confirmed
deadline expired
restructure
restructure in response to credit event
Amending(Restructuring)
agree price and confirm fees
schedule abandoned
schedule confirmed
Manually adjust cashflow schedule
Booking
A confirmation is generated for all events that result in a change to the trade. For example: a fixings confirmation on a fixing

Fig.9.

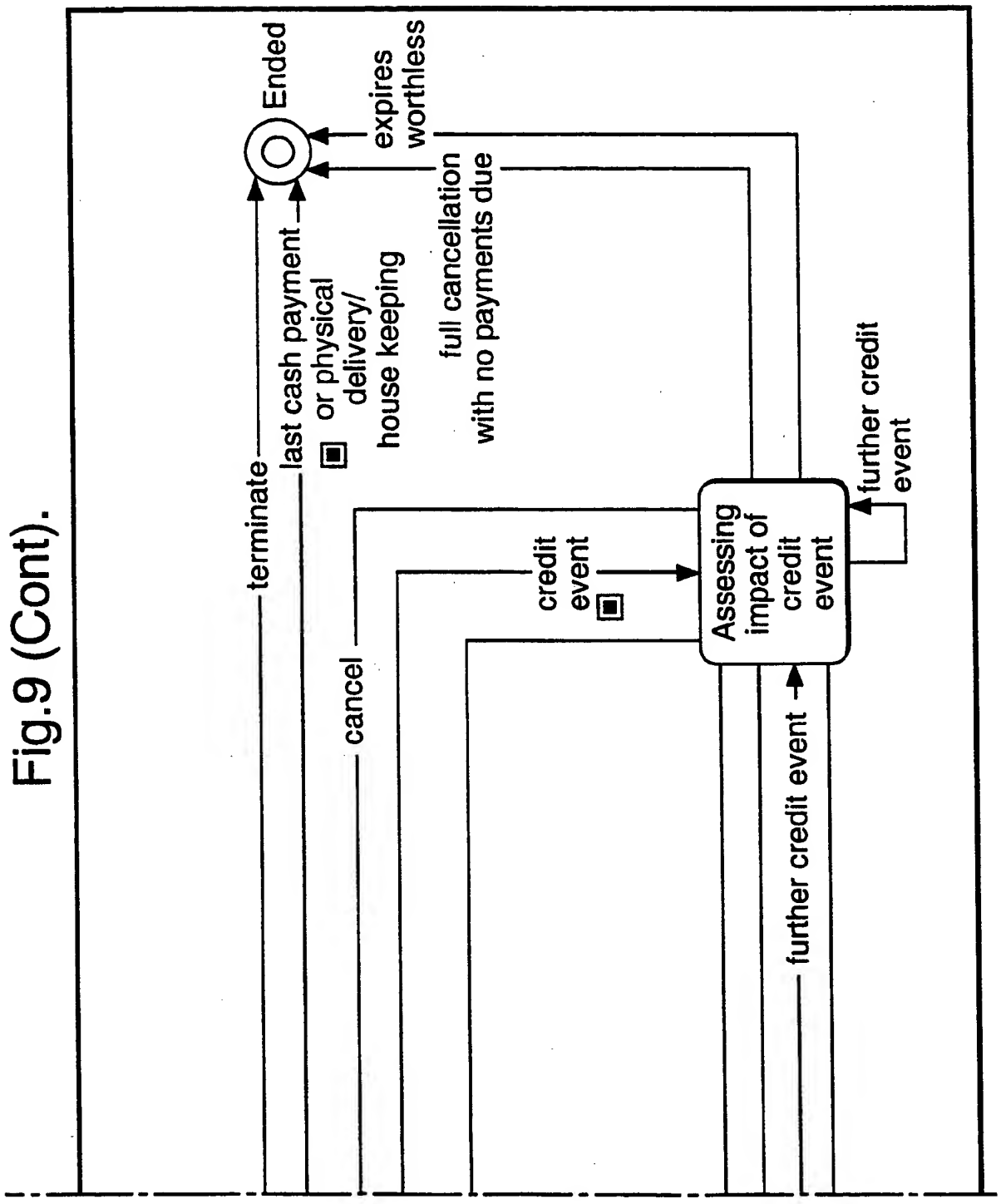

Ended
expires worthless
full cancellation with no payments due
last cash payment or physical delivery/ house keeping
terminate
cancel
credit event
Assessing impact of credit event
further credit event
further credit event

Fig.9 (Cont).

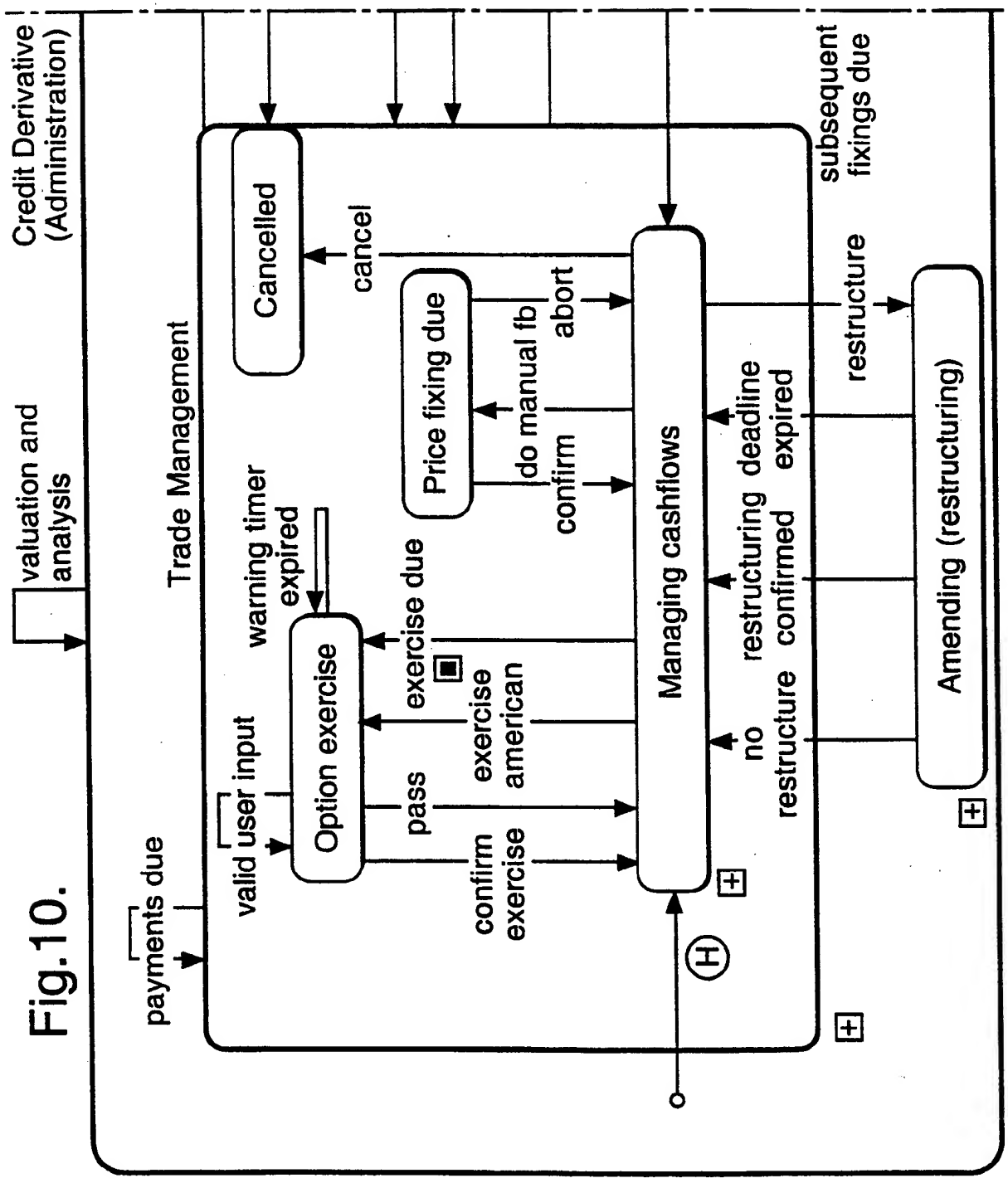

Credit Derivative (Administration)
valuation and analysis
Trade Management
payments due
valid user input
warning timer expired
Option exercise
exercise due
pass
exercise american
confirm exercise
Cancelled
cancel
Price fixing due
do manual fb
abort
confirm
Managing cashflows
no restructure
restructuring confirmed
deadline expired
restructure
subsequent fixings due
Amending (restructuring)
H

Fig.10.

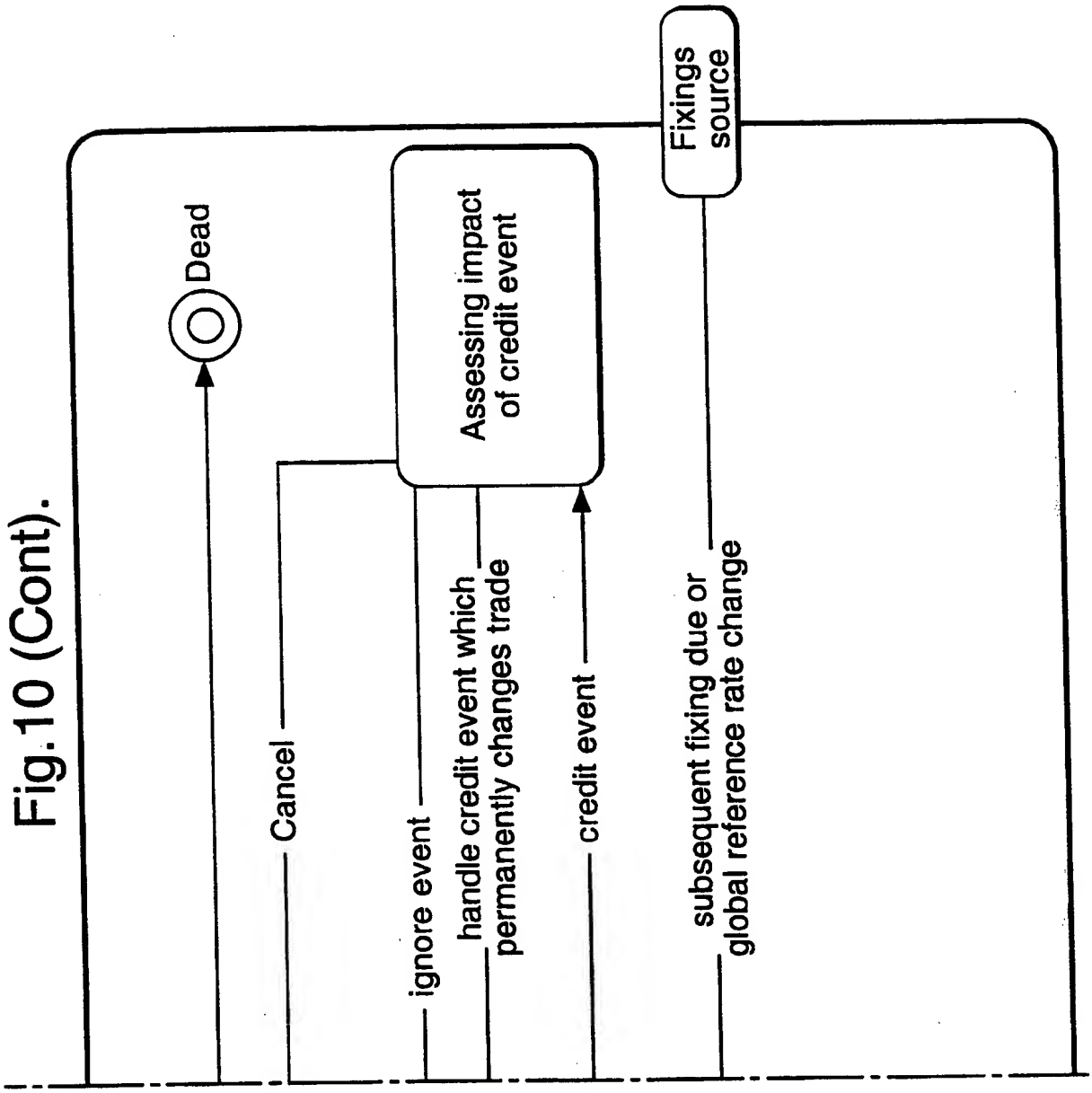
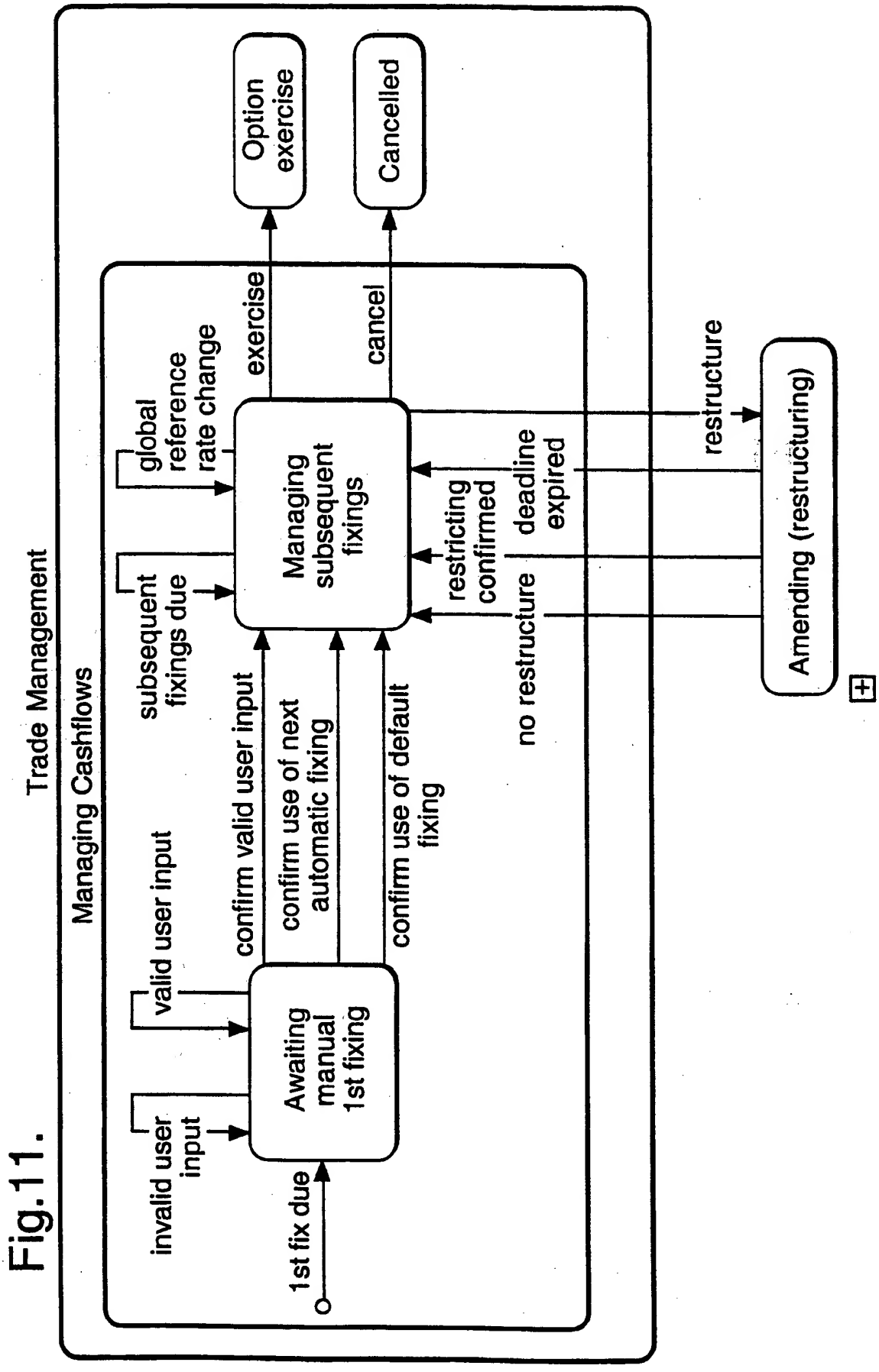

Fixings source
Assessing impact of credit event
Dead
Cancel
ignore event
handle credit event which permanently changes trade
credit event
subsequent fixing due or global reference rate change

Fig.10 (Cont).

Fig.11.


Trade Management
Managing Cashflows
1st fix due
invalid user input
valid user input
Awaiting manual 1st fixing
confirm valid user input
confirm use of next automatic fixing
confirm use of default fixing
subsequent fixings due
global reference rate change
Managing subsequent fixings
exercise
cancel
Option exercise
Cancelled
restructure
restricting confirmed
deadline expired
no restructure
Amending (restructuring)

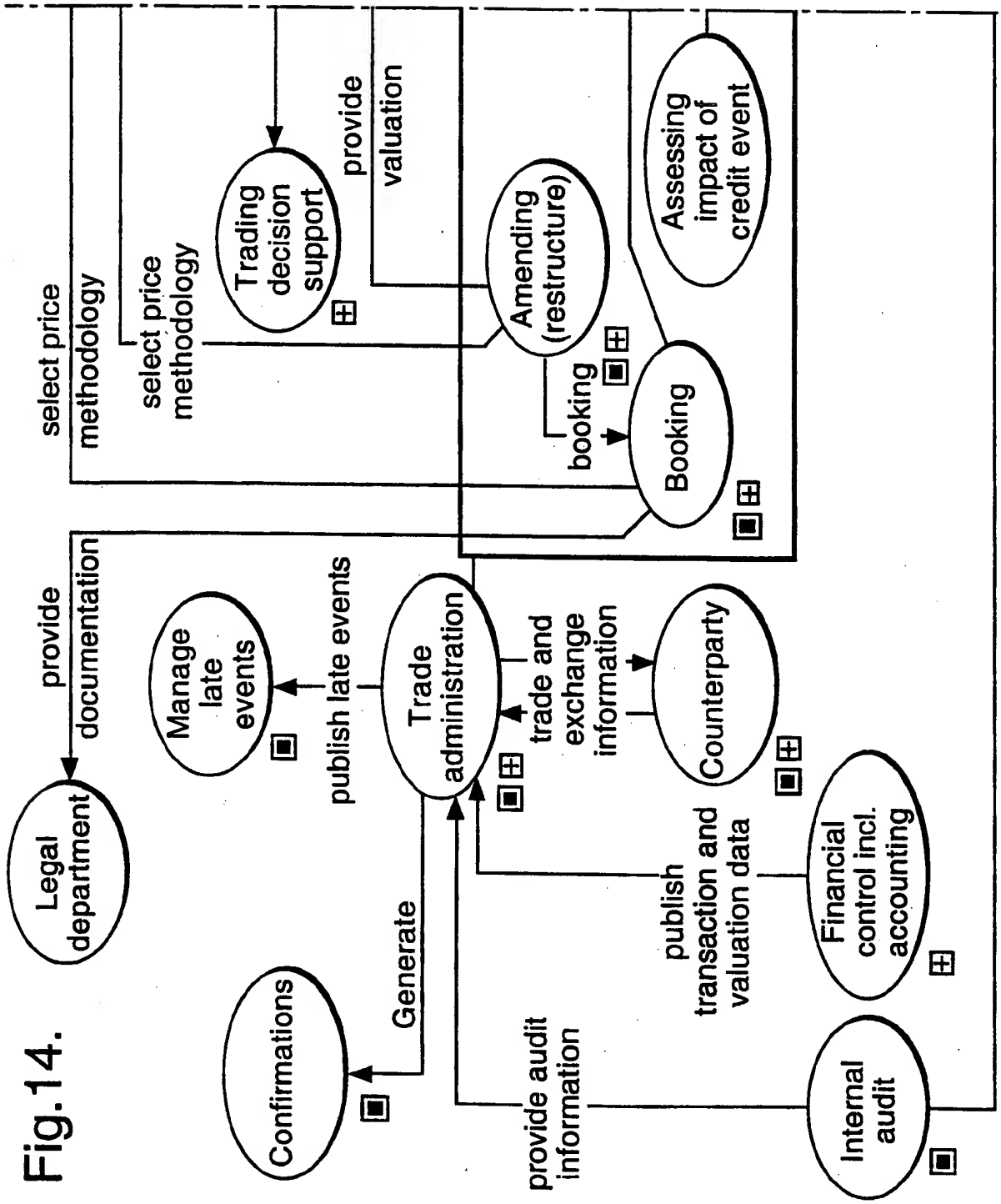
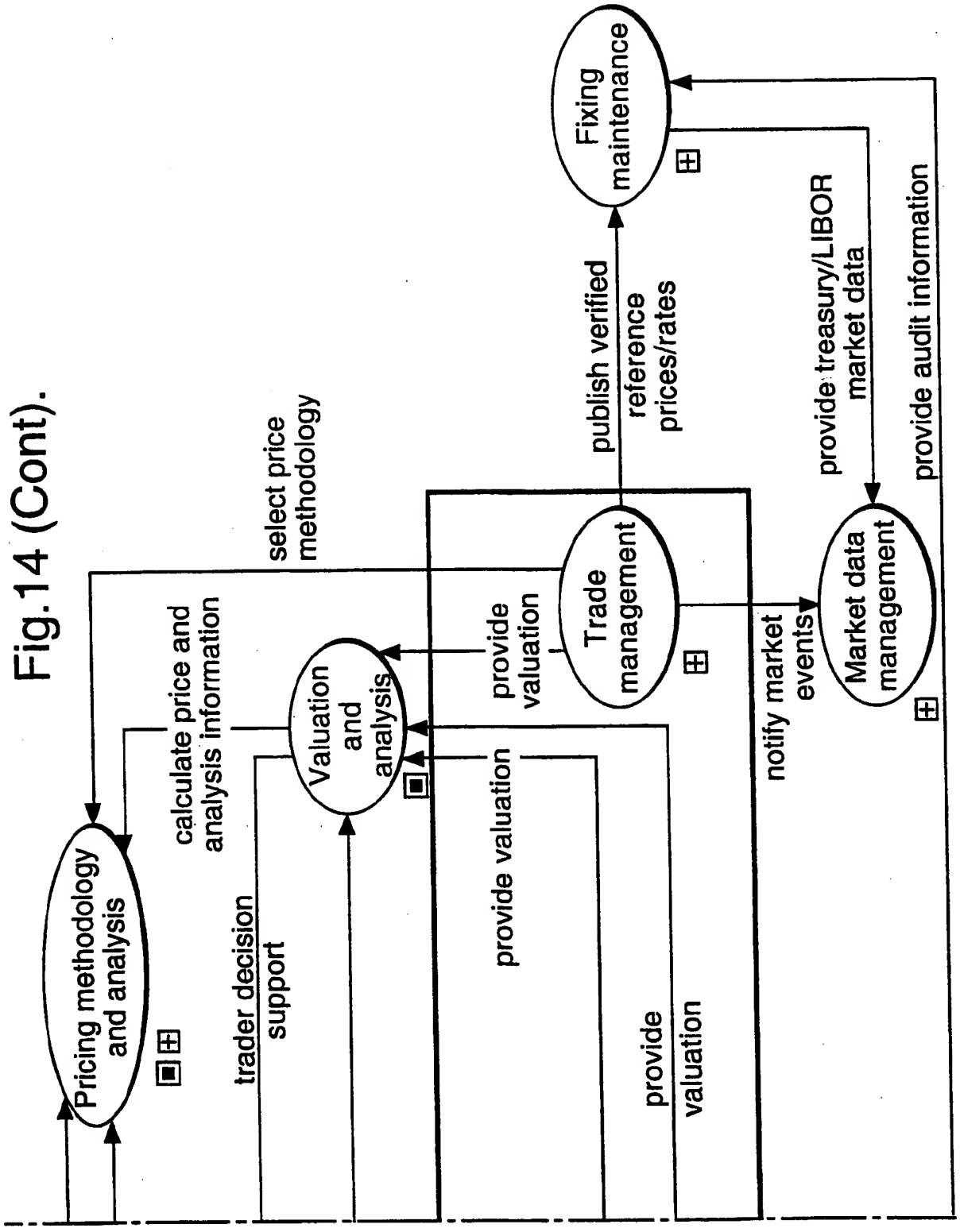

select price methodology
select price methodology
Trading decision support
provide valuation
Amending (restructure)
Assessing impact of credit event
booking
Booking
provide documentation
Manage late events
publish late events
Trade administration
trade and exchange information
Counterparty
Legal department
Generate
publish transaction and valuation data
Financial control incl. accounting
Confirmations
provide audit information
Internal audit

Fig.14.

Fig.14 (Cont).


Pricing methodology and analysis
Valuation and analysis
Trade management
Market data management
Fixing maintenance
select price methodology
calculate price and analysis information
trader decision support
provide valuation
provide valuation
provide valuation
notify market events
publish verified reference prices/rates
provide treasury/LIBOR market data
provide audit information

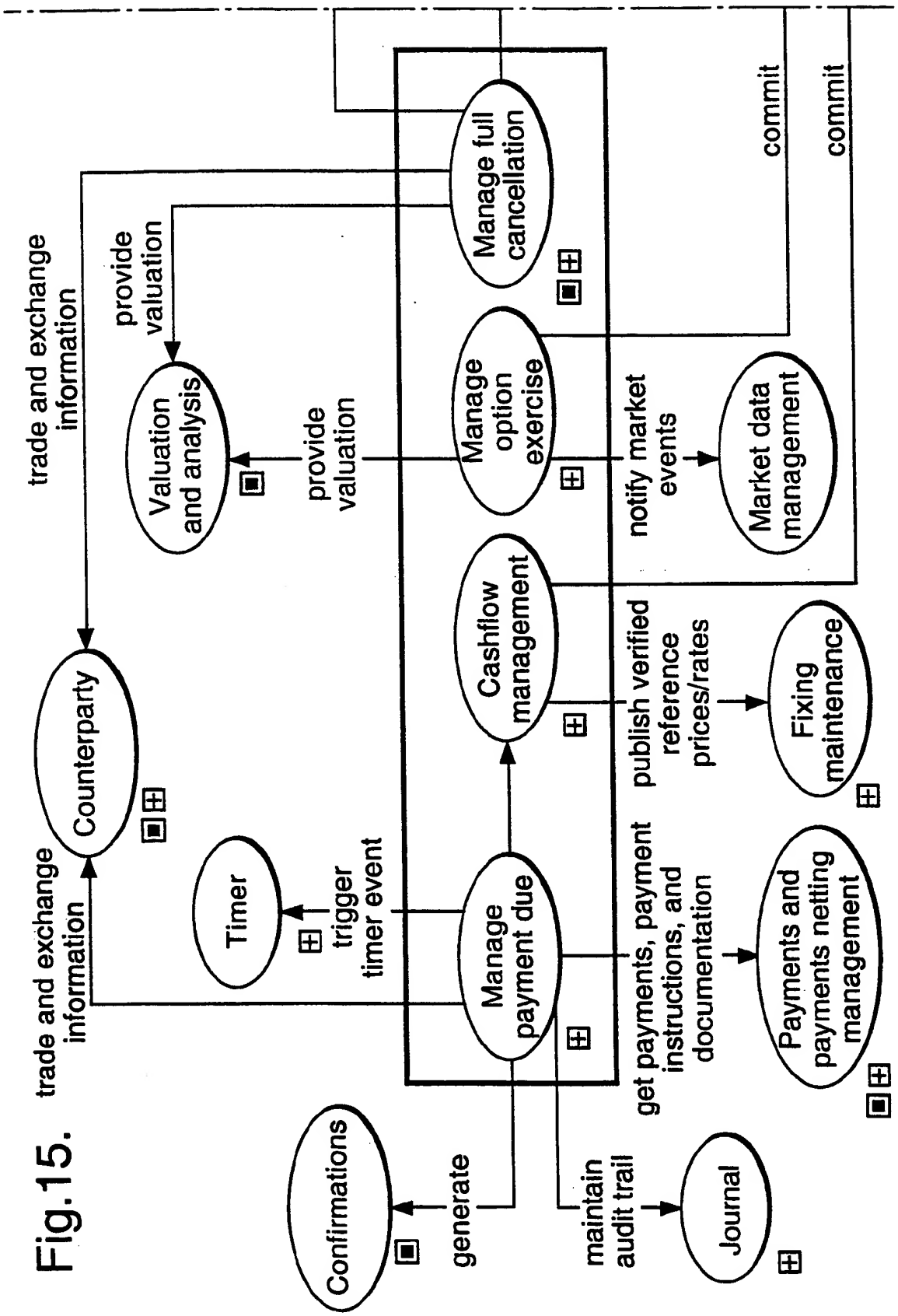
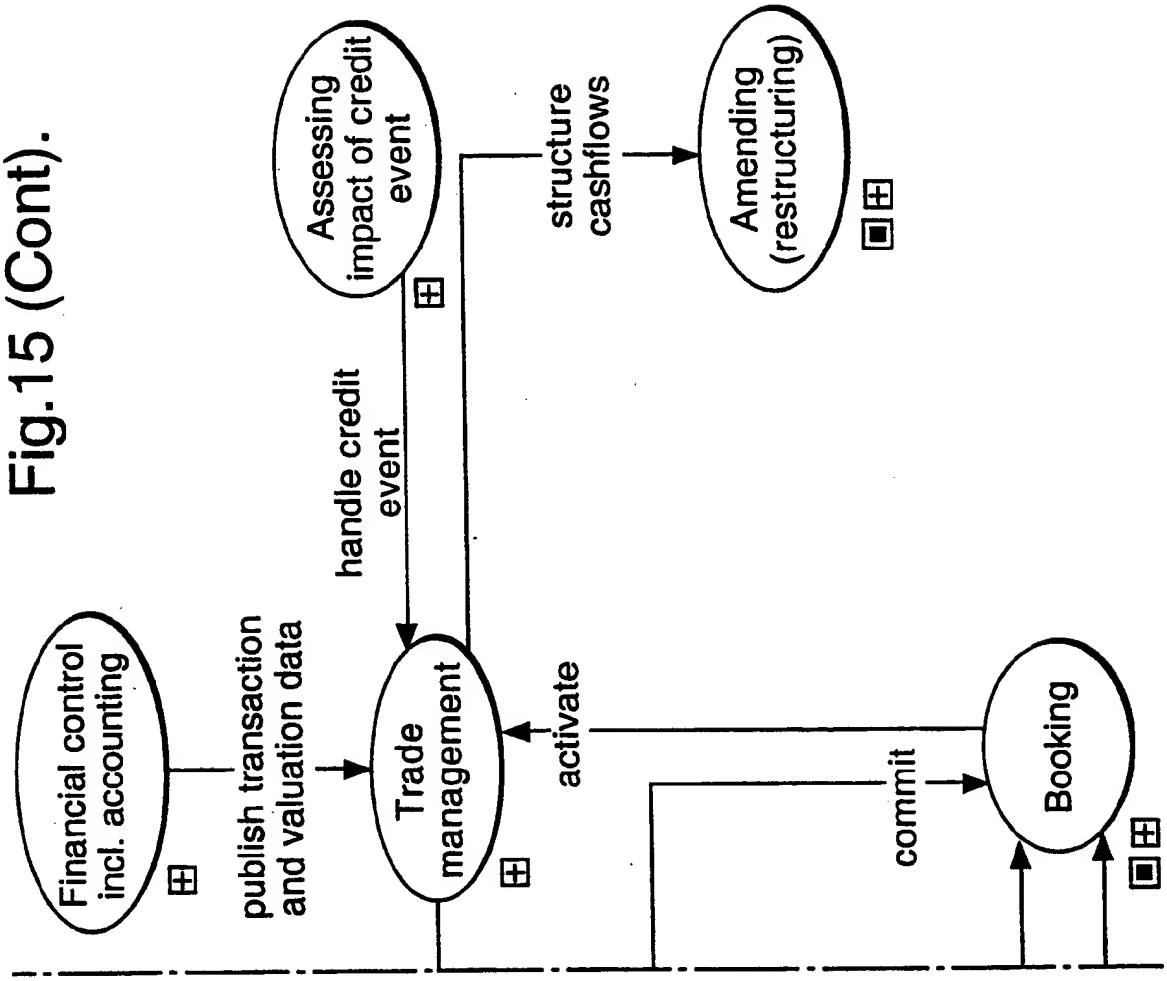

trade and exchange information
provide valuation
Valuation and analysis
provide valuation
Manage full cancellation
commit
commit
Manage option exercise
notify market events
Market data management
Cashflow management
publish verified reference prices/rates
Fixing maintenance
Counterparty
trade and exchange information
Timer
trigger timer event
Manage payment due
get payments, payment instructions, and documentation
Payments and payments netting management
Confirmations
generate
maintain audit trail
Journal

Fig.15.

Fig.15 (Cont).


Financial control incl. accounting
publish transaction and valuation data
Trade management
handle credit event
Assessing impact of credit event
structure cashflows
Amending (restructuring)
activate
commit
Booking

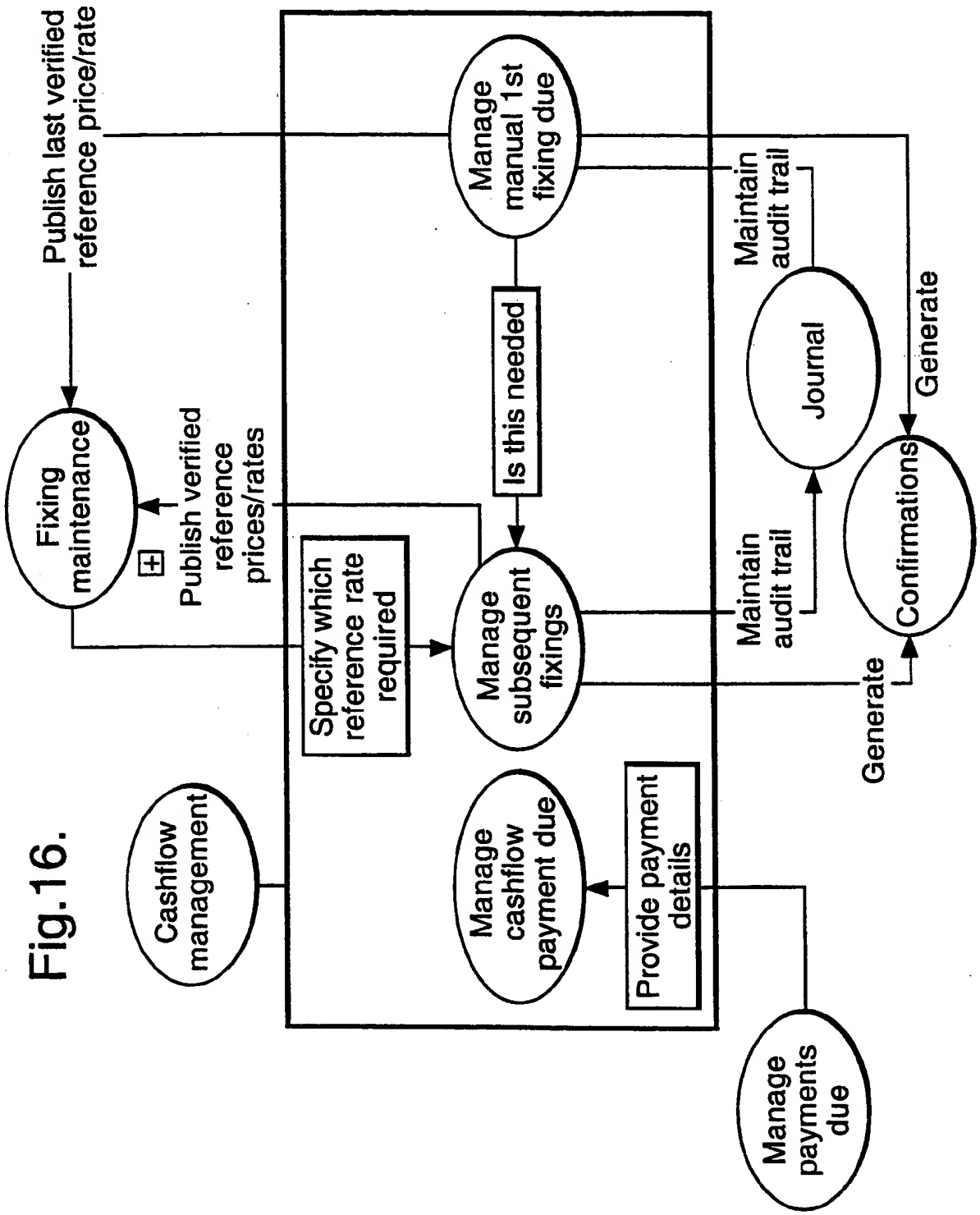

Publish last verified reference price/rate
Fixing maintenance
Publish verified reference prices/rates
Specify which reference rate required
Manage manual 1st fixing due
Is this needed
Manage subsequent fixings
Maintain audit trail
Journal
Maintain audit trail
Generate
Confirmations
Generate
Cashflow management
Manage cashflow payment due
Provide payment details
Manage payments due

Fig.16.

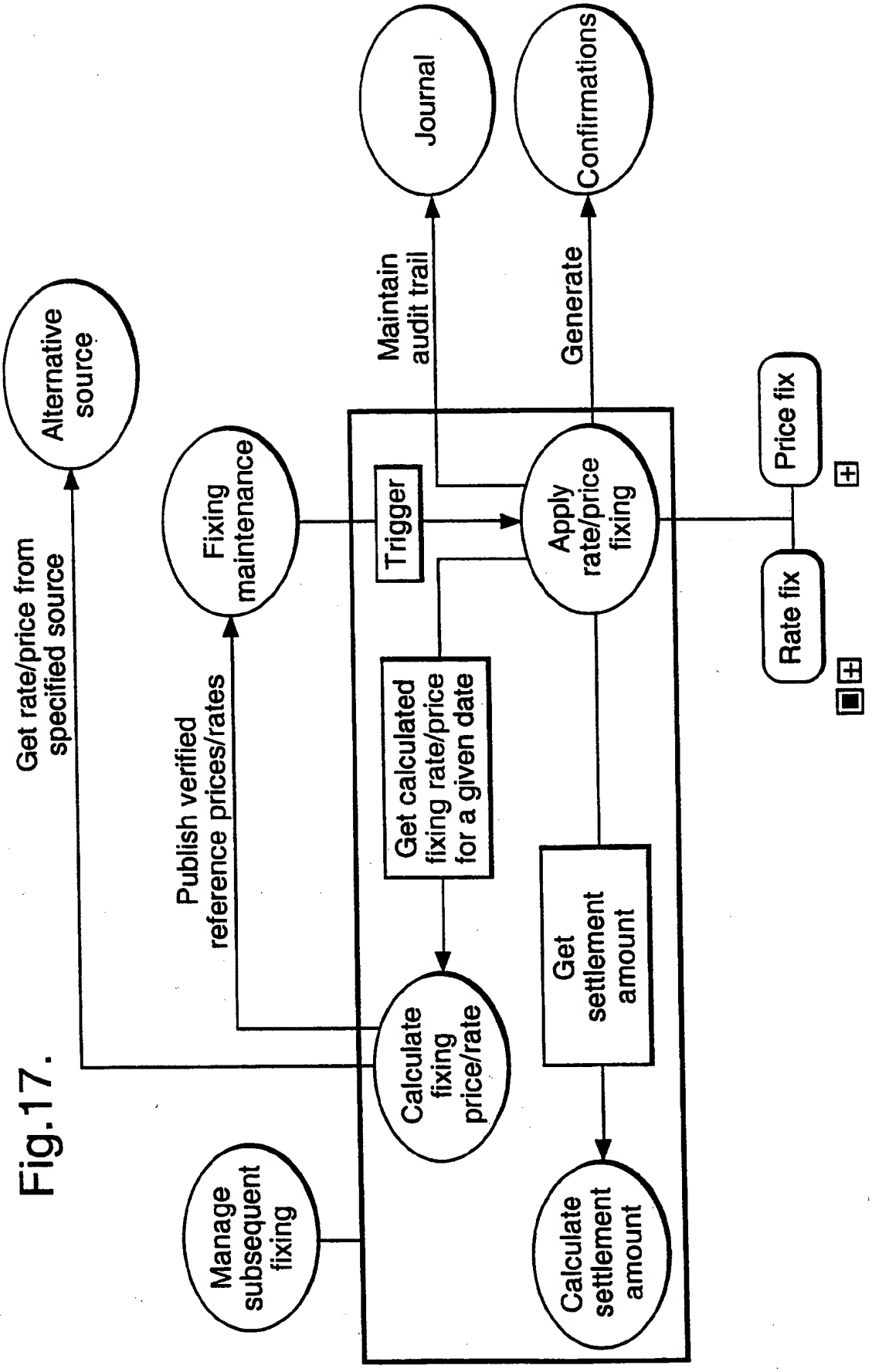

Alternative source
Get rate/price from specified source
Fixing maintenance
Publish verified reference prices/rates
Trigger
Get calculated fixing rate/price for a given date
Calculate fixing price/rate
Manage subsequent fixing
Maintain audit trail
Journal
Apply rate/price fixing
Generate
Confirmations
Get settlement amount
Calculate settlement amount
Price fix
Rate fix

Fig.17.

Fig.18.

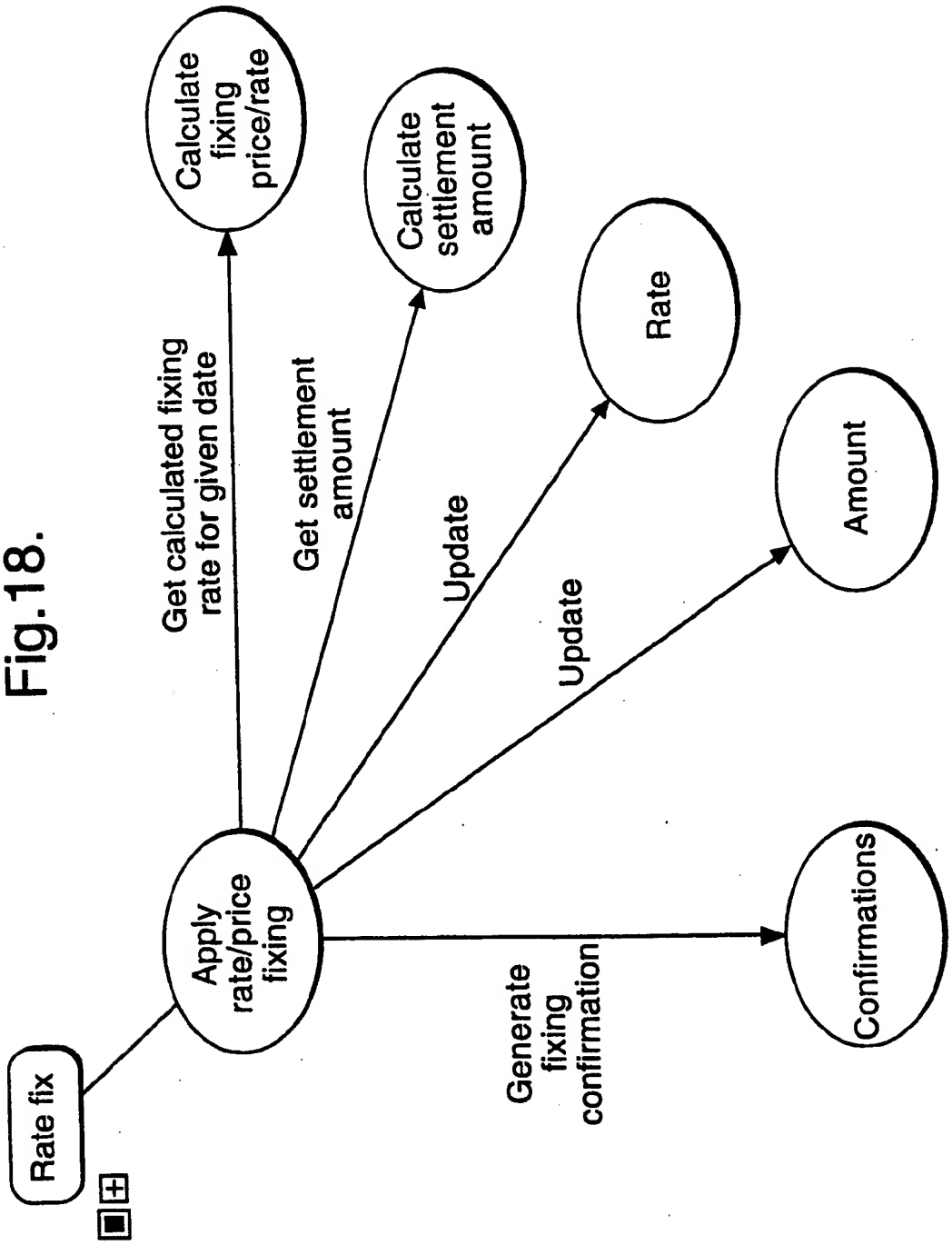
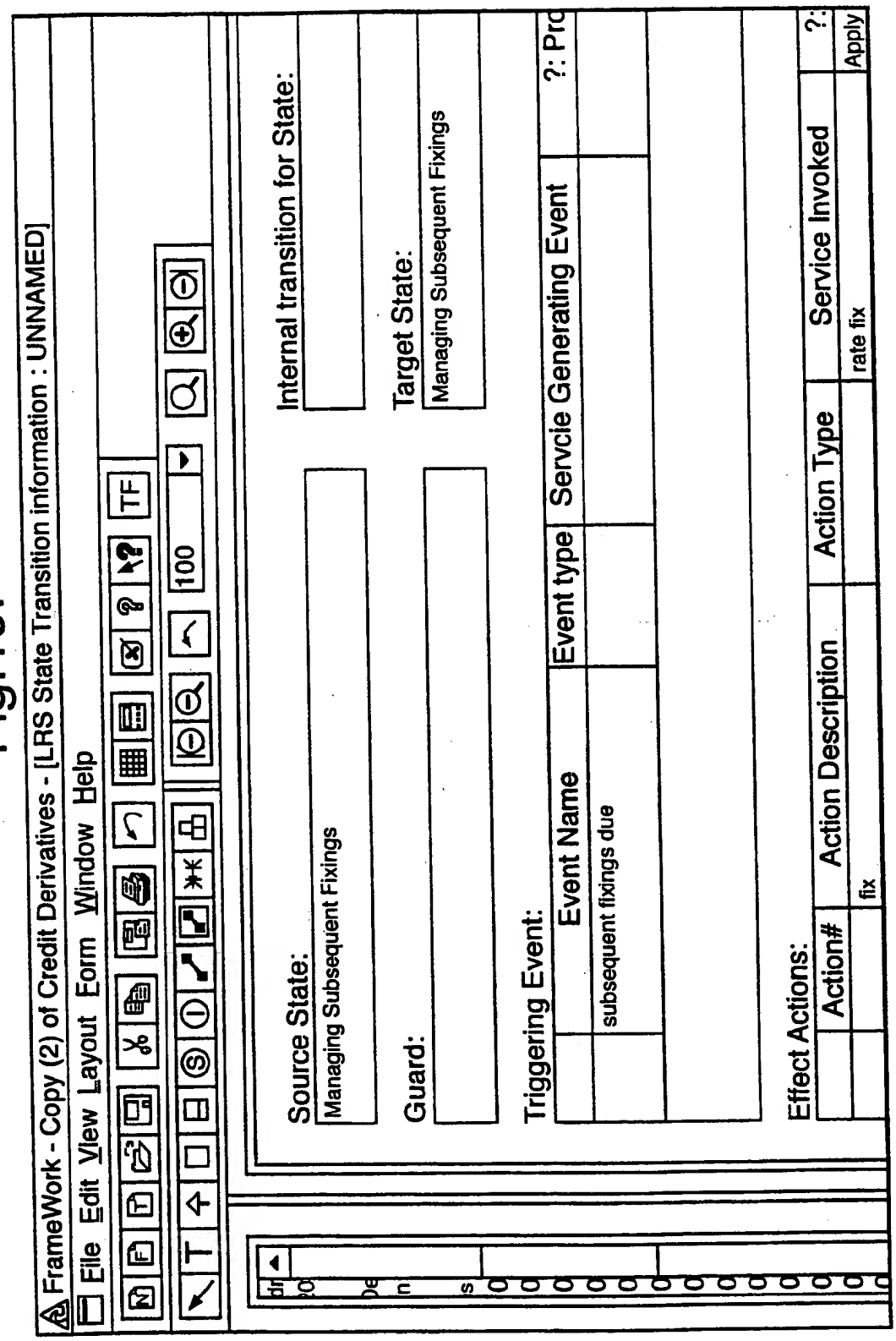

Rate fix
Apply rate/price fixing
Get calculated fixing rate for given date
Calculate fixing price/rate
Get settlement amount
Calculate settlement amount
Update
Rate
Update
Amount
Generate fixing confirmation
Confirmations

Fig.19.


FrameWork - Copy (2) of Credit Derivatives - [LRS State Transition information : UNNAMED]
File Edit View Layout Form Window Help
Source State:
Managing Subsequent Fixings
Internal transition for State:
Guard:
Target State:
Managing Subsequent Fixings
Triggering Event:
Event Name
Event type
Servcie Generating Event
?: Pro
subsequent fixings due
Effect Actions:
Action#
Action Description
Action Type
Service Invoked
?:
fix
rate fix
Apply

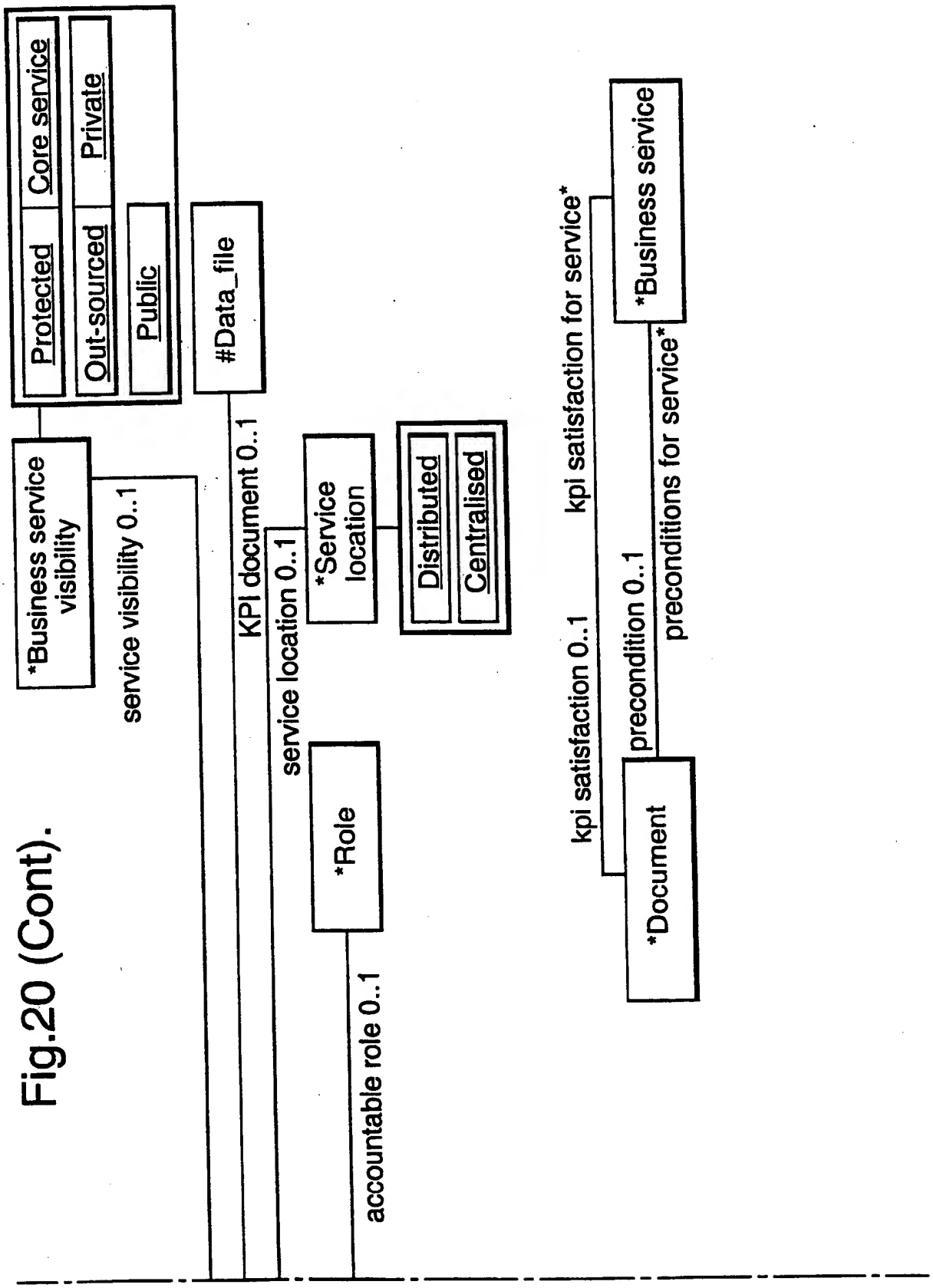
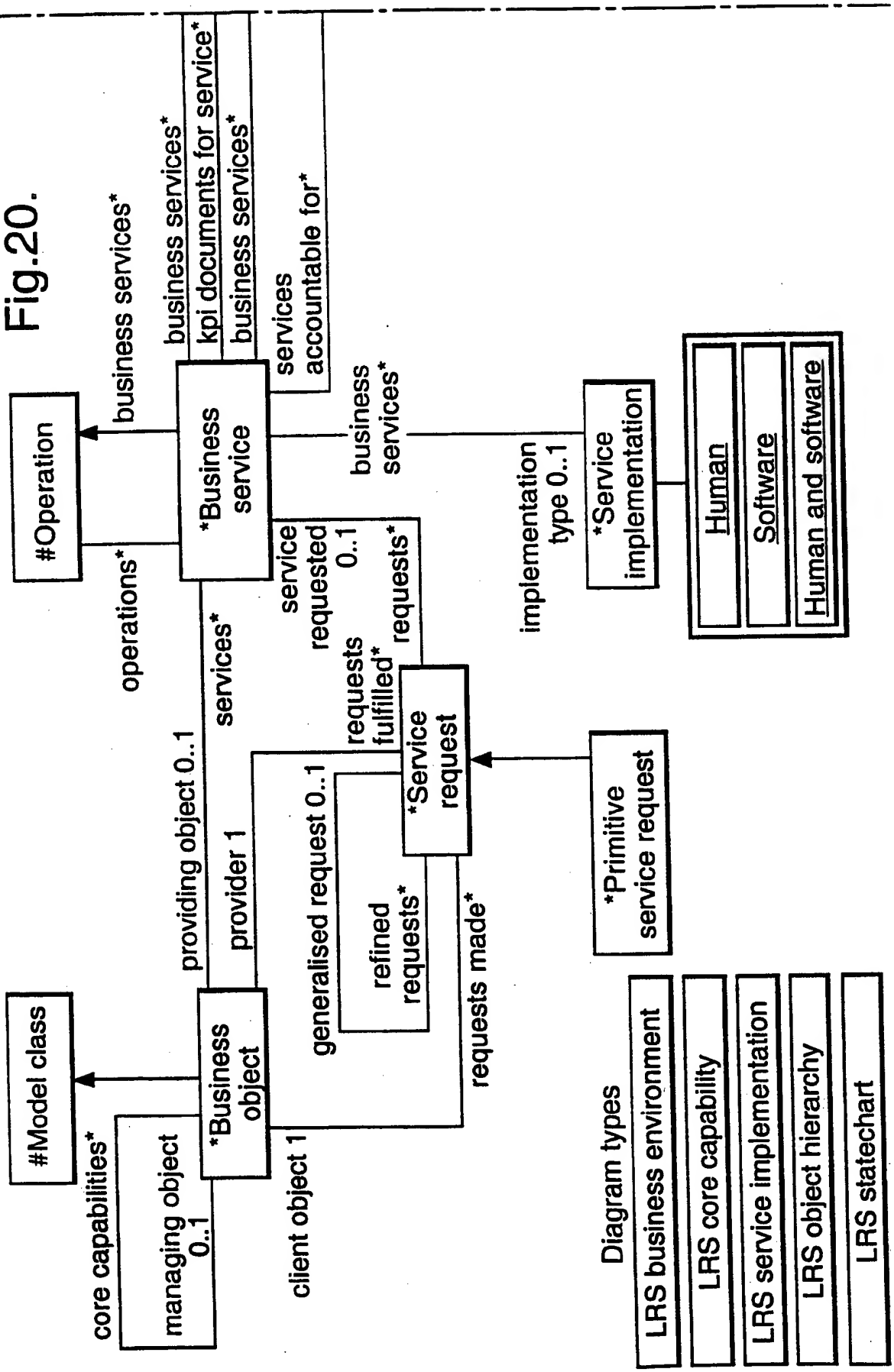

Protected
Core service
Out-sourced
Private
Public
*Business service visibility
service visibility 0..1
#Data_file
KPI document 0..1
*Service location
service location 0..1
Distributed
Centralised
*Role
accountable role 0..1
kpi satisfaction for service*
*Business service
kpi satisfaction 0..1
precondition 0..1
preconditions for service*
*Document

Fig.20 (Cont).

Fig.20.


#Operation
business services*
operations*
*Business service
business services*
kpi documents for service*
business services*
services accountable for*
service requested 0..1
requests fulfilled*
requests*
business services*
implementation type 0..1
*Service implementation
Human
Software
Human and software
providing object 0..1
services*
provider 1
generalised request 0..1
refined requests*
*Service request
requests made*
*Primitive service request
#Model class
core capabilities*
managing object 0..1
*Business object
client object 1
Diagram types
LRS business environment
LRS core capability
LRS service implementation
LRS object hierarchy
LRS statechart

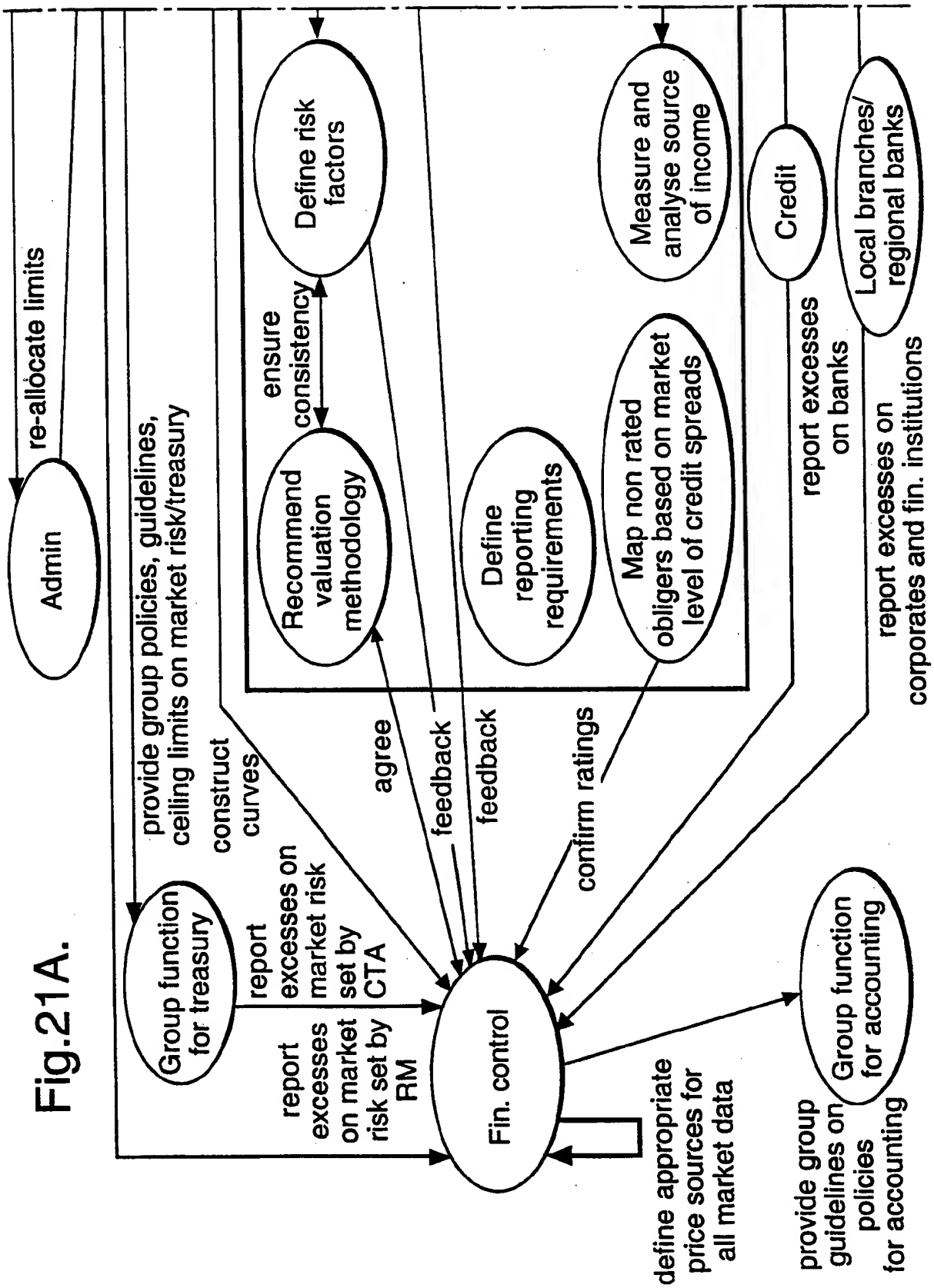

Admin
re-allocate limits
provide group policies, guidelines, ceiling limits on market risk/treasury
Group function for treasury
report excesses on market risk set by CTA
report excesses on market risk set by RM
construct curves
agree
feedback
feedback
confirm ratings
Fin. control
define appropriate price sources for all market data
Define risk factors
ensure consistency
Recommend valuation methodology
Define reporting requirements
Map non rated obligers based on market level of credit spreads
Measure and analyse source of income
Credit
report excesses on banks
Local branches/ regional banks
report excesses on corporates and fin. institutions
Group function for accounting
provide group guidelines on policies for accounting

Fig.21A.

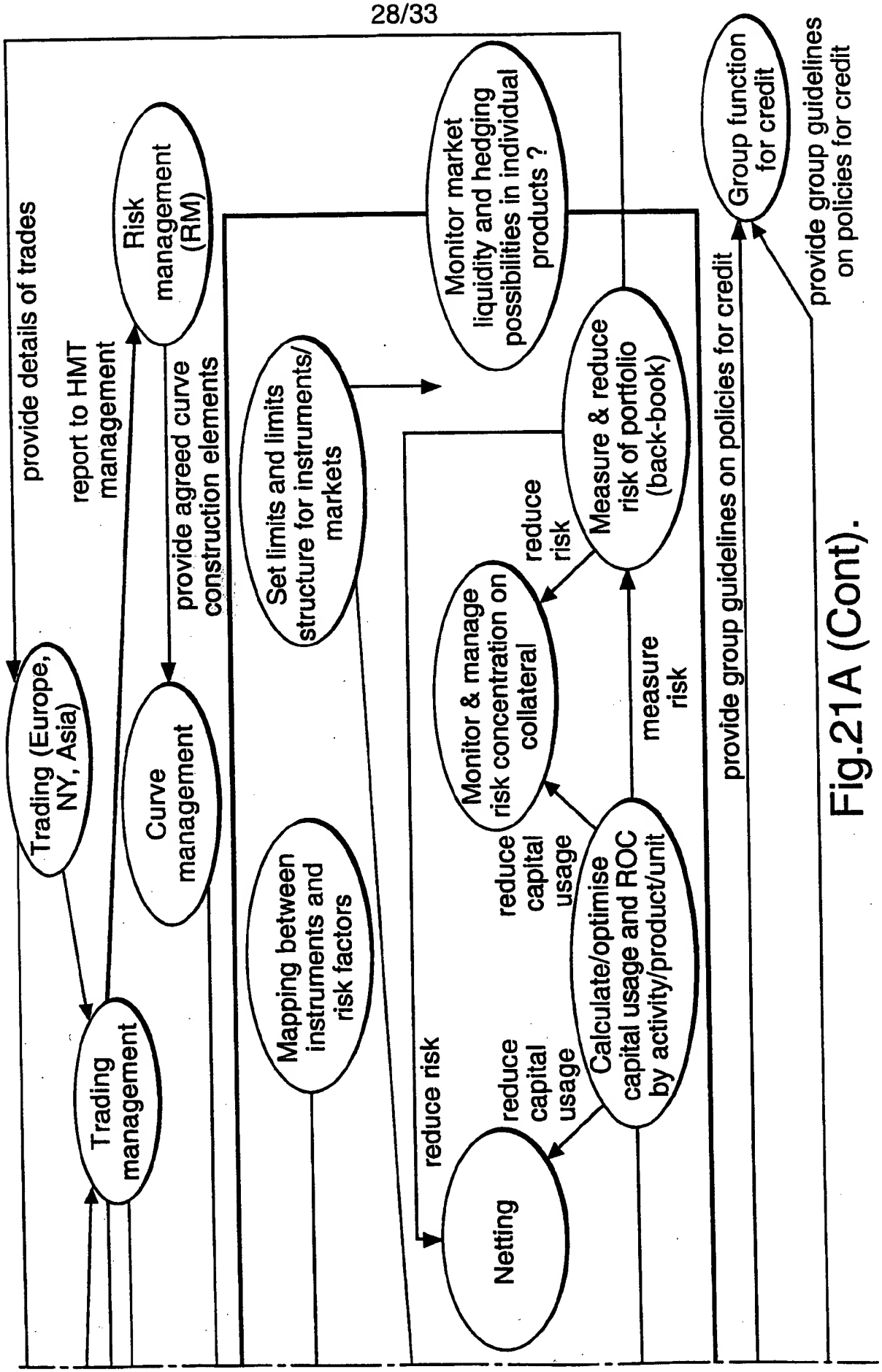
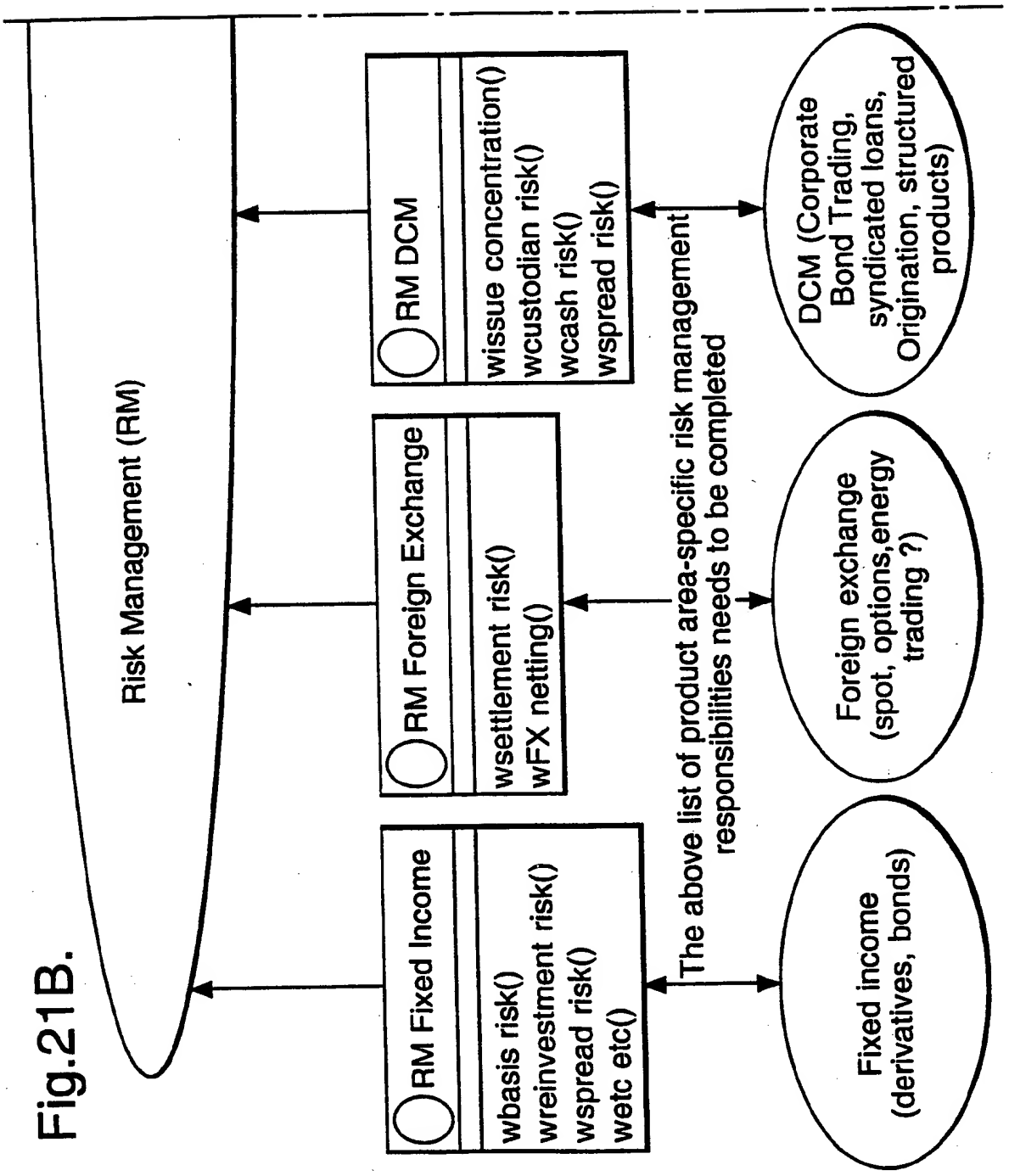

provide details of trades
report to HMT management
provide agreed curve construction elements
Risk management (RM)
Trading (Europe, NY, Asia)
Curve management
Trading management
Mapping between instruments and risk factors
Set limits and limits structure for instruments/ markets
Monitor market liquidity and hedging possibilities in individual products ?
Measure & reduce risk of portfolio (back-book)
reduce risk
Monitor & manage risk concentration on collateral
measure risk
reduce capital usage
Calculate/optimise capital usage and ROC by activity/product/unit
reduce risk
reduce capital usage
Netting
provide group guidelines on policies for credit
Group function for credit
provide group guidelines on policies for credit

Fig.21A (Cont).

Fig.21B.


Risk Management (RM)
RM Fixed Income
wbasis risk()
wreinvestment risk()
wspread risk()
wetc etc()
RM Foreign Exchange
wsettlement risk()
wFX netting()
RM DCM
wissue concentration()
wcustodian risk()
wcash risk()
wspread risk()
The above list of product area-specific risk management responsibilities needs to be completed
Fixed income (derivatives, bonds)
Foreign exchange (spot, options,energy trading ?)
DCM (Corporate Bond Trading, syndicated loans, Origination, structured products)

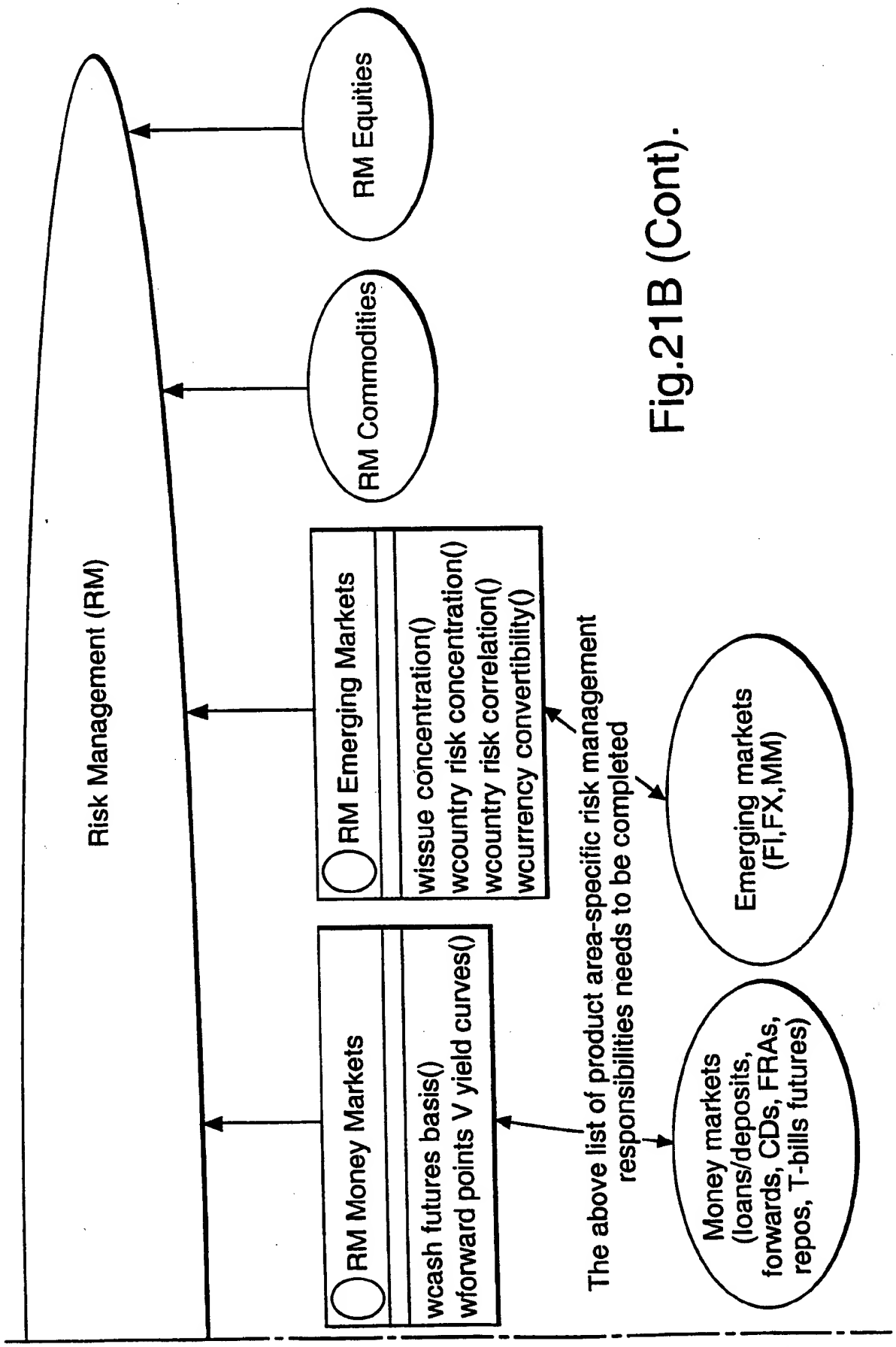
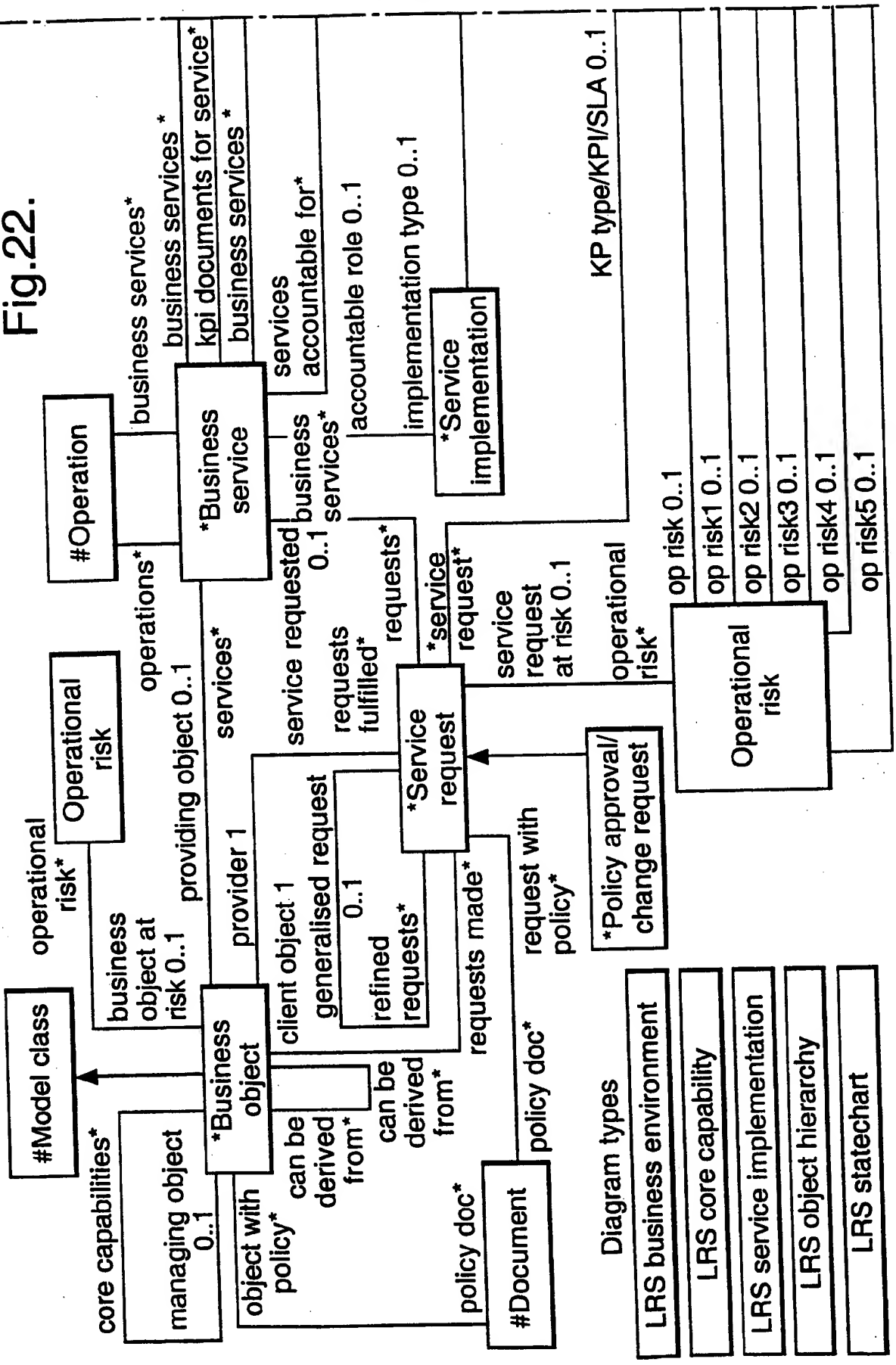

Risk Management (RM)
RM Equities
RM Commodities
RM Emerging Markets
wissue concentration()
wcountry risk concentration()
wcountry risk correlation()
wcurrency convertibility()
RM Money Markets
wcash futures basis()
wforward points V yield curves()
The above list of product area-specific risk management responsibilities needs to be completed
Emerging markets (FI,FX,MM)
Money markets (loans/deposits, forwards, CDs, FRAs, repos, T-bills futures)

Fig.21B (Cont).

Fig.22.


#Model class
core capabilities*
managing object 0..1
*Business object
object with policy*
can be derived from*
can be derived from*
policy doc*
#Document
policy doc*
request with policy*
requests made*
client object 1
generalised request 0..1
refined requests*
provider 1
business object at risk 0..1
operational risk*
Operational risk
providing object 0..1
operations*
#Operation
business services*
*Business service
business services *
kpi documents for service*
business services *
service requested 0..1
requests fulfilled*
*Service request
service request
service request at risk 0..1
*Policy approval/ change request
services*
business services*
services accountable for*
accountable role 0..1
implementation type 0..1
*Service implementation
operational risk*
Operational risk
KP type/KPI/SLA 0..1
op risk 0..1
op risk1 0..1
op risk2 0..1
op risk3 0..1
op risk4 0..1
op risk5 0..1
Diagram types
LRS business environment
LRS core capability
LRS service implementation
LRS object hierarchy
LRS statechart

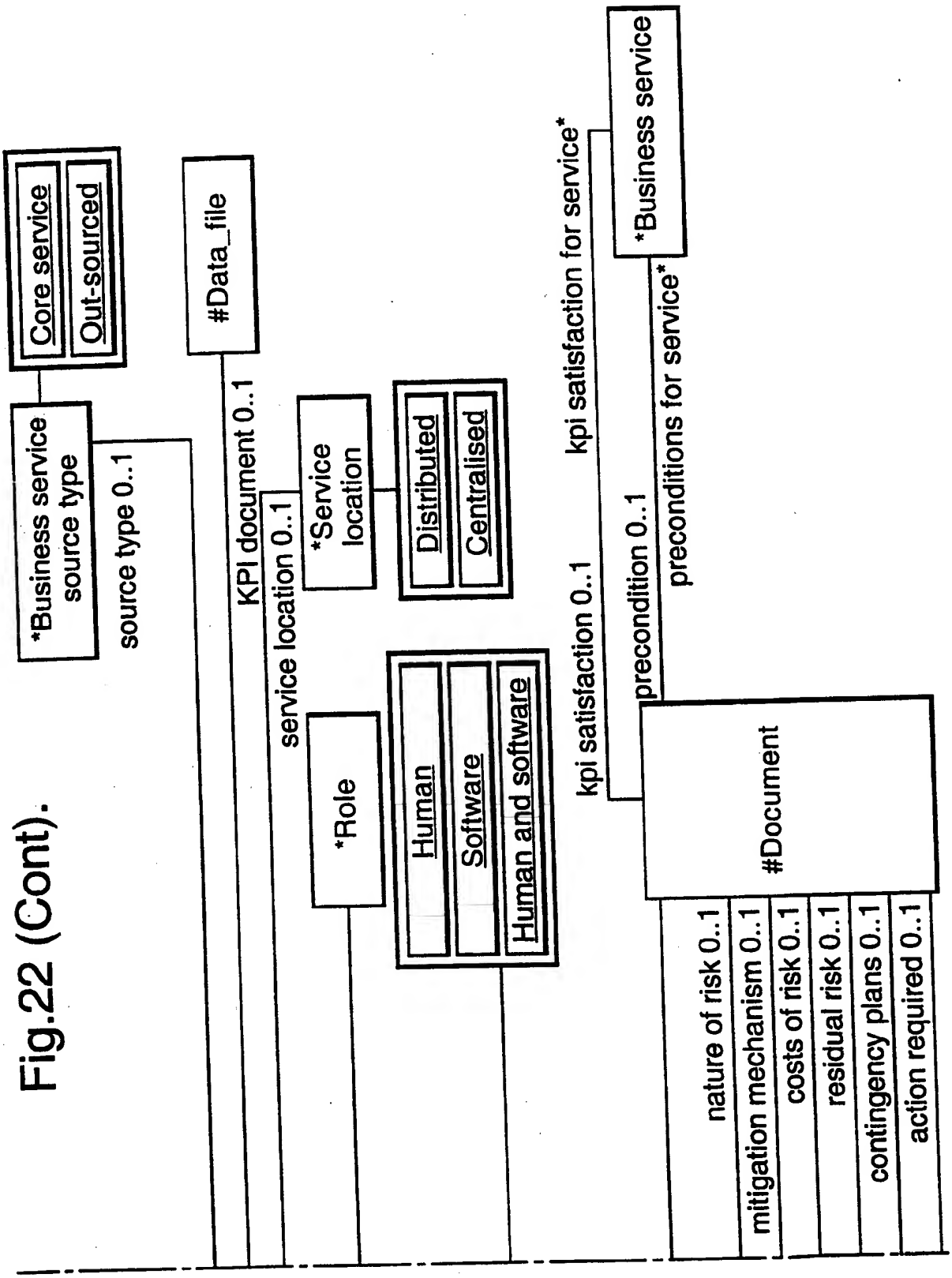
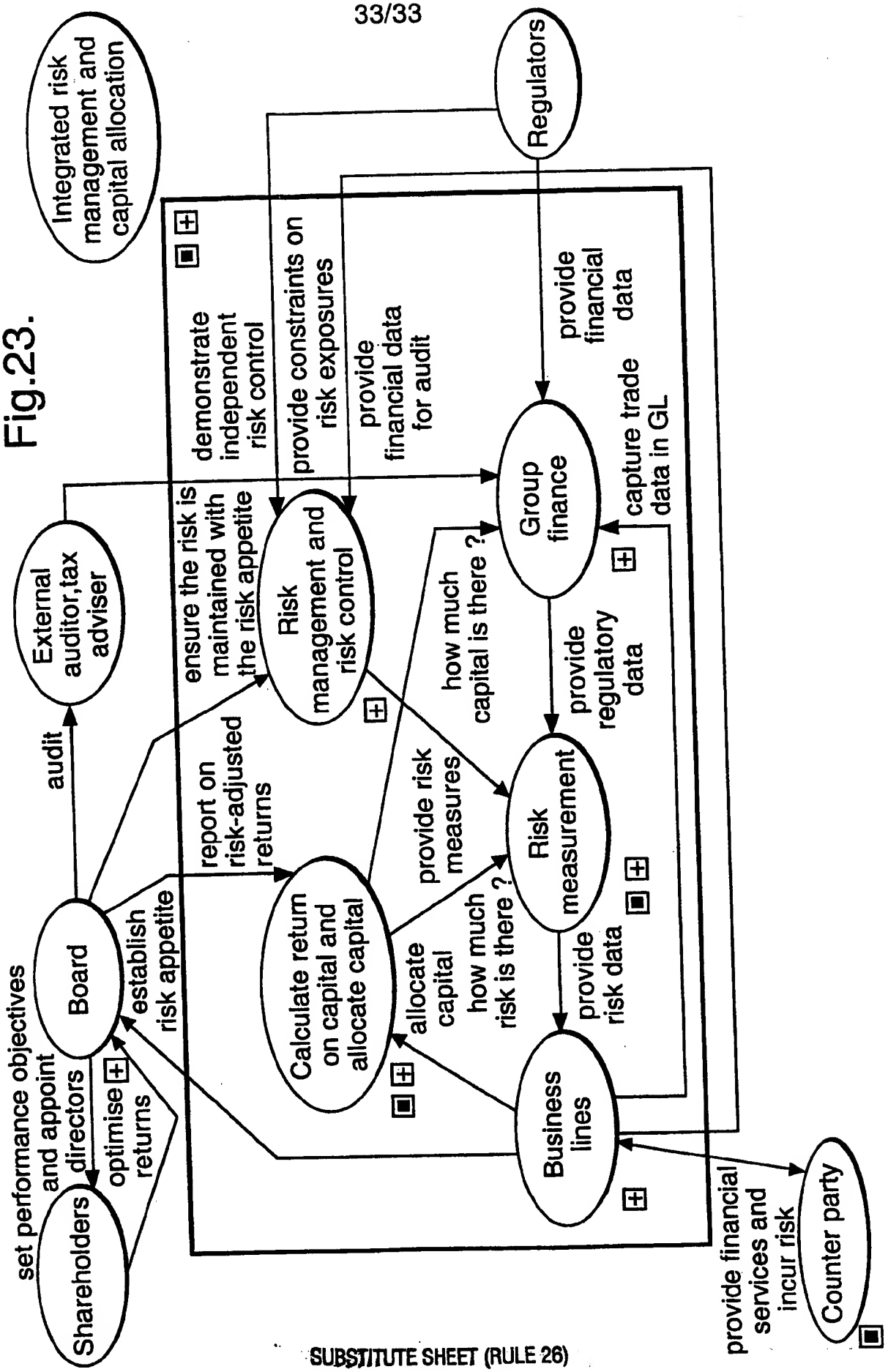

Core service
Out-sourced
*Business service source type
source type 0..1
#Data_file
KPI document 0..1
service location 0..1
*Service location
Distributed
Centralised
*Role
Human
Software
Human and software
kpi satisfaction 0..1
kpi satisfaction for service*
*Business service
precondition 0..1
preconditions for service*
#Document
nature of risk 0..1
mitigation mechanism 0..1
costs of risk 0..1
residual risk 0..1
contingency plans 0..1
action required 0..1

Fig.22 (Cont).

Fig.23.


Integrated risk management and capital allocation
Regulators
demonstrate independent risk control
provide constraints on risk exposures
provide financial data for audit
provide financial data
capture trade data in GL
External auditor,tax adviser
ensure the risk is maintained with the risk appetite
Risk management and risk control
Group finance
how much capital is there ?
provide regulatory data
audit
report on risk-adjusted returns
provide risk measures
Risk measurement
Board
establish risk appetite
Calculate return on capital and allocate capital
allocate capital
how much risk is there ?
provide risk data
set performance objectives and appoint directors
optimise returns
Business lines
Shareholders
provide financial services and incur risk
Counter party

THIS PAGE BLANK (USPTO)

INTERNATIONAL SEARCH REPORT

Int ional Application No
PCT/GB 00/00297

A. CLASSIFICATION OF SUBJECT MATTER
IPC 7 G06F17/60

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
IPC 7 G06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	BUSTARD D W ET AL: "Integrating soft systems and object-oriented analysis" PROCEEDINGS OF THE SECOND INTERNATIONAL CONFERENCE ON REQUIREMENTS ENGINEERING (CAT. NO.96TB100037), PROCEEDINGS OF THE SECOND INTERNATIONAL CONFERENCE ON REQUIREMENTS ENGINEERING, COLORADO SPRINGS, CO, USA, 15-18 APRIL 1996, pages 52-59, XP002120073 1996, Los Alamitos, CA, USA, IEEE Comput. Soc. Press, USA ISBN: 0-8186-7252-8	1-3,17, 20-24
A	page 52, column 1 -page 57, column 1	4-16,18, 19
X	DE 196 12 688 A (IBM) 2 October 1997 (1997-10-02) column 2, line 7 -column 4, line 62 column 5, line 31 -column 9, line 15	1-3,17, 20,21

-/--

[X] Further documents are listed in the continuation of box C.

[X] Patent family members are listed in annex.

* Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
24 March 2000	31/03/2000

Name and mailing address of the ISA	Authorized officer
European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Tx. 31 651 epo nl, Fax: (+31-70) 340-3016	Bowler, A

INTERNATIONAL SEARCH REPORT

International Application No
PCT/GB 00/00297

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 5 446 885 A (MOORE ALLAN R ET AL) 29 August 1995 (1995-08-29)	25
A	column 16, line 27 -column 17, line 2 column 3, line 8 -column 5, line 27; figure 13	26-28
A	MERTINS K ET AL: "A tool for object-oriented modelling and analysis of business processes" COMPUTERS IN INDUSTRY, vol. 33, no. 2-3, 1 September 1997 (1997-09-01), page 345-356 XP004097799 ISSN: 0166-3615 page 346, column 1 -page 351, column 2; figures 1-6	1-24
A	US 5 535 389 A (ELDER DAVID B ET AL) 9 July 1996 (1996-07-09)	29-32

INTERNATIONAL SEARCH REPORT

Information on patent family members

Int tional Application No
PCT/GB 00/00297

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
DE 19612688 A	02-10-1997	NONE	
US 5446885 A	29-08-1995	US 5630127 A	13-05-1997
US 5535389 A	09-07-1996	JP 7129401 A	19-05-1995

This Page is Inserted by IFW Indexing and Scanning Operations and is not part of the Official Record

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ BLACK BORDERS
- ☐ IMAGE CUT OFF AT TOP, BOTTOM OR SIDES
- ☒ FADED TEXT OR DRAWING
- ☐ BLURRED OR ILLEGIBLE TEXT OR DRAWING
- ☐ SKEWED/SLANTED IMAGES
- ☐ COLOR OR BLACK AND WHITE PHOTOGRAPHS
- ☐ GRAY SCALE DOCUMENTS
- ☐ LINES OR MARKS ON ORIGINAL DOCUMENT
- ☐ REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY
- ☐ OTHER: _____

IMAGES ARE BEST AVAILABLE COPY.
As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.